

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ
ІНСТИТУТ**

Кафедра математичного моделювання та аналізу даних

«До захисту допущено»
Б.о. завідувача кафедри
_____ I.M. Терещенко
«___» ____ 2023 р.

**Дипломна робота
на здобуття ступеня бакалавра**

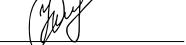
зі спеціальності: 113 Прикладна математика
на тему: «Розробка системи виявлення аномалій в
багатовимірних даних з використанням глибинного навчання»

Виконав: студент 4 курсу, групи ФІ-92
Кириленко Павло Олександрович



Керівник: асистент каф. ММАД О.А. Яворський

Рецензент: старший викладач Ю.В. Наконечна



Засвідчую, що у цій дипломній
роботі немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ
ІНСТИТУТ**

Кафедра математичного моделювання та аналізу даних

Рівень вищої освіти — перший (бакалаврський)
Спеціальність (освітня програма) — 113 Прикладна математика,
ОПП «Математичні методи моделювання, розпізнавання образів та
безпеки даних»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ І.М. Терещенко
«___» _____ 2023 р.

**ЗАВДАННЯ
на дипломну роботу**

Студент: Кириленко Павло Олександрович

1. Тема роботи: «*Розробка системи виявлення аномалій в багатовимірних даних з використанням глибинного навчання*»,

керівник: асистент каф. ММАД О.А. Яворський,

затверджені наказом по університету №____ від «___» _____ 2023 р.

2. Термін подання студентом роботи: «___» _____ 2023 р.

3. Вихідні дані до роботи: публіковані джерела за тематикою дослідження.

4. Зміст роботи: аналіз проблем, що виникають при розв'язку задачі виявлення аномалій в багатовимірних часових рядах за допомогою сучасних нейромережевих архітектур та розробка методів для розв'язання цієї задачі.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація доповіді.

6. Дата видачі завдання: 15 листопада 2022 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	15-30 листопада 2022 р.	Виконано
2	Огляд сучасних методів розв'язання задачі виявлення аномалій в багатовимірних часових рядах	Січень-лютий 2023 р.	Виконано
3	Дослідження метрик та методологій оцінювання для задачі виявлення аномалій	15 січня - 10 лютого 2023 р	Виконано
4	Дослідження синхронізованих часових рядів	Лютий 2023 р.	Виконано
5	Реалізація моделей для виявлення аномалій в багатовимірних часових рядах	10 - 20 квітня 2023 р	Виконано
6	Аналіз результатів реалізованих моделей	10 - 20 травня 2023 р.	Виконано
7	Оформлення дипломної роботи	Травень 2023 р.	Виконано

Студент

П.О. Кириленко

Керівник

О.А. Яворський



РЕФЕРАТ

Кваліфікаційна робота містить: 57 стор., 24 рисунки, 4 таблиць, 16 джерел.

Через значний ріст даних, які отримуються з датчиків як за обсягами (кількість записів), так і за розмірністю (кількість ознак) традиційні статистичні методи аналізу простору станів або контролювані інструменти навчання вже не є ефективними. Тому постає проблема розробки нових методів.

Мета роботи полягає в розробці системи виявлення аномалій в багатовимірних даних з використанням глибинного навчання. Ця система включає предобробку даних, побудову моделей, теоретична та практична оцінка моделей, порівняння моделей на різних датасетах.

У ході виконання роботи було проведені експерименти над моделями, які належать різним підходам вирішення цієї задачі. В цій роботі розглядаються RNN LSTM модель на основі прогнозування часових рядів та модель RANCoders на основі реконструкції, а також їхні модифікації SynLSTM, RANSynCoders (що використовують інформацію про те, що ряди є асинхронними). В цій роботі було доведена ефективність запропонованих моделей для застосування в серверних системах. Що може свідчити про те, що моделі глибинного навчання є спроможними розв'язувати такі задачі на високому рівні.

**ВИЯВЛЕННЯ АНОМАЛІЙ, БАГАТОВИМІРНІ ДАНІ,
СИНХРОНІЗАЦІЯ ЧАСОВИХ РЯДІВ, RNN, LSTM, АЕ,
RANSYNCODERS, НАВЧАННЯ БЕЗ УЧИТЕЛЯ**

ABSTRACT

The qualification work contains: 57 pages, 24 figures, 4 tables, 16 sources.

Due to the significant growth of sensor data both in volume (number of records) and dimensionality (number of features), traditional statistical methods of state space analysis or supervised learning tools are no longer effective. Therefore, the problem of developing new methods arises.

The purpose of the work is to develop a system for detecting anomalies in multidimensional data using deep learning. This system includes data preprocessing, model building, theoretical and practical evaluation of models, comparison of models on different datasets.

In the course of the work, experiments were conducted with models belonging to different approaches to solving this problem. This paper examines the RNN LSTM model based on time series forecasting and the RANCoders model based on reconstruction, as well as their modifications SynLSTM, RANSynCoders (which use the information that the series are asynchronous). This work proved the effectiveness of the proposed models for use in server systems. Which may indicate that deep learning models are capable of solving such problems very well.

ANOMALY DETECTION, MULTIDIMENSIONAL DATA, TIME SERIES SYNCHRONIZATION, RNN, LSTM, AE, RANSYNCODERS, UNSUPERVISED LEARNING

ЗМІСТ

Перелік умовних позначень, скорочень і термінів	7
Вступ.....	8
1 Огляд сучасних методів виявлення аномалій в багатовимірних даних	10
1.1 Сучасні нейромережеві моделі для виявлення аномалій на основі підходу прогнозування	10
1.2 Сучасні нейромережеві моделі для виявлення аномалій на основі підходу реконструкції	14
1.3 Нейронні мережі на основі реконструкції з використанням багатосигнальної синхронності	17
Висновки до розділу 1	22
2 Огляд алгоритмів пошуку порогового значення та метрики оцінки якості моделей.....	24
2.1 Метрики якості моделей	24
2.2 Алгоритм сліпого пошуку порогового значення	27
2.3 Алгоритм пошуку найкращого теоретичного порогового значення	30
Висновки до розділу 2	32
3 Реалізація та аналіз моделей	34
3.1 Підготовка даних	34
3.2 Реалізація RNN LSTM та SynLSTM моделей	36
3.3 Реалізація AE RANCoders та RANSynCoders моделей	42
3.4 Порівняння результатів якості виявлення аномалій в даних	47
Висновки до розділу 3	49
Висновки	52
Перелік посилань	55

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ANN – artificial neural networks, штучна нейронна мережа

RNN – recurrent neural networks, рекурентна нейронна мережа

AE – AutoEncoder, автокодувальник

LSTM – long short-term memory, довга короткочасна пам'ять

SynLSTM – LSTM модель з вбудованим шаром синхронізації часових рядів

RANCoders – Recurrent Autoencoder Network Coders, Рекурентний мережевий AE

RANSynCoders – RANCoders з вбудованим шаром синхронізації часових рядів

PCA – principal component analysis, метод головних компонент

ВСТУП

Актуальність дослідження. Завдяки розвитку технологій і поширенню датчиків і пристройів моніторингу тепер ми маємо доступ до величезних обсягів даних. Прикладом галузей, що генерують велику кількість інформації є такі галузі, як фінанси, кібербезпека та промисловий моніторинг. Одним із видом такої інформації є часові ряди, а саме багатовимірні часові ряди. Через це зростає потреба у виявленні аномалій, щоб запобігти шахрайству, порушенням безпеки або збоям обладнання. Традиційні методи такі як методи на основі перевірки порівняння з нормою (Rule-based anomaly detection), статистичні методи, аналізу часових рядів або застосування машинного навчання на основі моделей для класифікації або кластеризації можуть бути неспроможними ефективно впоратися з високою розмірністю та кількістю вхідних даних, що потребує розробки спеціалізованих систем виявлення аномалій. Наразі сучасні загрози та кібератаки характеризуються своєю витонченістю, складністю та новизною. Системи виявлення аномалій, обладнані для роботи з цими характеристиками, можуть відігравати важливу роль у виявленні та пом'якшенні нових загроз, захисті систем і мереж, а також забезпечені цілісності та безпеки критично важливих програм та інфраструктури.

Метою дослідження є розробка системи для виявлення аномалій в багатовимірних часових даних з використанням глибинного навчання.

Задача дослідження полягає у дослідженні методів глибинного навчання для розв'язання задачі пошуку аномалій в часових рядах. Для розв'язання задачі необхідно вирішити такі **завдання**:

- 1) оглянути сучасні методи розв'язання задачі пошуку аномалій;
- 2) дослідити алгоритми вибору порогового значення для розмежування аномальних та нормальніх даних та метрики оцінки якості моделей;

3) побудувати відповідні моделі на основі прогнозування і реконструкції даних та проаналізувати, і оцінити результати моделей.

Об'єктом дослідження є проблема виявлення аномалій в багатовимірних часових рядах на попередньо нерозмічених даних.

Предметом дослідження питання розробки ефективних моделей глибинного навчання для пошуку аномалій в багатовимірних часових рядах.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: методи побудови та навчання глибоких нейронних мереж, методи математичної статистики та математичного аналізу.

Наукова новизна отриманих результатів полягає в розробці моделей на основі прогнозування та реконструкції для пошуку аномалій в багатовимірних часових рядах з використанням будованого шару для синхронізації часових рядів та оцінці ефективності застосування моделей для серверних систем, що використовуються в промислових цілях.

Практичне значення результатів полягає у покращенні надійності систем та попередження аварій. Розроблена система виявлення аномалій може бути застосована для моніторингу та контролю різних промислових систем. Це допоможе вчасно виявляти аномальні події, уникати відмов та запобігати серйозним аваріям.

Апробація результатів та публікації. Частина даної роботи була представлена на ХХІ Науково-практичній конференції студентів, аспірантів та молодих вчених в секції «Математичного моделювання та аналізу даних» (11 травня 2023 р., м. Київ).

1 ОГЛЯД СУЧАСНИХ МЕТОДІВ ВИЯВЛЕННЯ АНОМАЛІЙ В БАГАТОВИМІРНИХ ДАНИХ

У першому розділі даної роботи буде проведено огляд нейромережевих методів, які застосовують для розв'язання задачі виявлення аномалій.

1.1 Сучасні нейромережеві моделі для виявлення аномалій на основі підходу прогнозування

Означення 1.1. Аномалія в даних часового ряду належати до точки даних або послідовності точок даних, яка значно відхиляється від очікуваної або нормальній поведінки часового ряду. Він представляє незвичайну або аномальну модель, тенденцію або подію, яка виділяється серед звичайних моделей, що спостерігаються в даних. Аномалії можуть виникати внаслідок різних факторів, таких як помилки в зборі даних, несправності обладнання, системні збої, шахрайські дії або значні зміни в базовому процесі генерації даних. Виявлення аномалій у даних часових рядів має вирішальне значення для визначення потенційних проблем, викидів або критичних подій, які потребують подального дослідження чи втручання.

Підхід на основі прогнозування використовує навчену модель для прогнозування точки або послідовності на основі точки або останнього вікна. Щоб визначити, наскільки аномальними є вхідні значення, прогнозовані значення порівнюються з їхніми фактичними значеннями. Відхилення від фактичних значень яке більше попередньо обраного граничного значення вважаються аномальними значеннями. Більшість методів прогнозування використовують ковзне вікно для прогнозування однієї точки за раз. Щоб визначити аномальну поведінку, вони

використовують предиктор для моделювання нормальної поведінки. Варто зазначити, що в деяких попередніх роботах, таких як [1], помилка передбачення використовується як кількісна характеристика новизни, а не оцінка аномалії.

Далі варто розглянути більш детально приклад моделі з підходом на основі прогнозування.

Означення 1.2 (англ. Artificial Neural Networks, ANNs). Штучні нейронні мережі — це комп’ютерні моделі, які намагаються імітувати роботу нервової системи людини для розв’язання задач штучного інтелекту. Вони використовують навчання на основі даних, щоб автоматично вивчати складні шаблони і залежності у вхідних даних і здійснювати високорівневий аналіз.

ANNs складаються зі шарів нейронів, де кожен нейрон пов’язаний з нейронами на попередньому та наступному шарах. ANNs здатні працювати зі статичними даними та використовують пряму подачу сигналів (feedforward) від вхідного до вихідного шару. Основна ідея ANNs полягає в тому, що нейрони в кожному шарі обчислюють зважену суму своїх вхідних сигналів та застосовують нелінійну активаційну функцію до цієї суми.

Означення 1.3 (англ. Recurrent Neural Networks , RNN). Рекурентна нейронна мережа — це тип нейронної мережі, спеціально призначений для роботи з послідовними даними. Вона використовує рекурентні зв’язки для передачі інформації від попередніх кроків обробки до поточного.

Означення 1.4 (англ. Long Short-Term Memory, LSTM). Довга короткочасна пам’ять — це спеціальний тип RNN, який був розроблений для ефективного моделювання довгострікових залежностей в послідовних даних. LSTM є покращенням стандартних RNN та дозволяє зберігати та використовувати інформацію з різних моментів часу в довгостріковій перспективі.

Традиційні ANNs розглядають входи та виходи як окремі значення.

RNN також враховують інформацію з попередніх вхідних даних і, отже, можуть ефективно обробляти послідовності даних. У той час як стандартні RNN застосовують поточкову нелінійність до афінного перетворення входів і рекурентних одиниць. У LSTM замість одного нейрона в рекурентному шарі використовується спеціальна структура, яка включає три важливі компоненти: входи (input gate), вихідні дані (output gate) та забування (forget gate). Кожен з цих елементів має свою вагу, яка регулює потік інформації [2]. Ці структури надають можливість LSTM вирішувати, коли та яку інформацію зберігати, яку інформацію забути та яку інформацію використовувати для вихідних значень. Це дає змогу моделі ефективно працювати з довгостроковими залежностями в послідовних даних.

Блоки LSTM (Рис. 1.1) складаються з векторів попереднього стану c'_{t-1} , попереднього стану комірки h'_{t-1} , вхідного стану комірки i'_t та вихідного стану комірки o'_t .

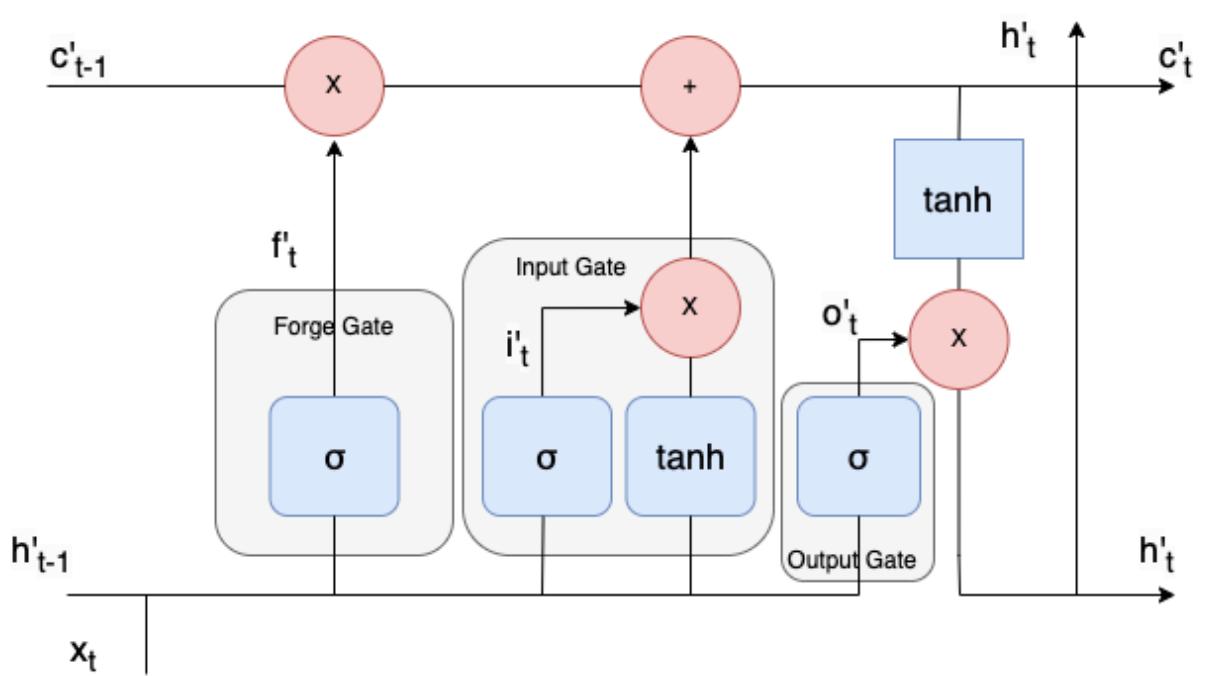


Рисунок 1.1 – LSTM модель

Рівняння для прямого проходу LSTM такі:

$$f_t = (\mathbf{W}_f \cdot x_t + \mathbf{U}_f \cdot h_{t-1} + b_f) \quad (1.1)$$

$$i_t = (\mathbf{W}_i \cdot x_t + \mathbf{U}_i \cdot h_{t-1} + b_i) \quad (1.2)$$

$$o_t = (\mathbf{W}_o \cdot x_t + \mathbf{U}_o \cdot h_{t-1} + b_o) \quad (1.3)$$

$$o_t = f_t c_{t-1} + i_t \tanh(\mathbf{W}_o \cdot x_t + \mathbf{U}_o \cdot h_{t-1} + b_o) \quad (1.4)$$

$$h_t = \tanh(c_t) o_t \quad (1.5)$$

де σ , \mathbf{W} , \mathbf{U} , b відповідно позначають функцію активації (сигмоїда в цій роботі), вхідні ваги, рекурентні ваги та зміщення.

Використання LSTM для аналізу багатовимірних часових даних дозволяє моделі здійснювати прогнозування, класифікацію та виявлення аномалій з врахуванням контексту, що може покращити точність та надійність аналітичних рішень в цьому контексті.

Для виявлення аномалій необхідно навчити модель на даних, що не містять аномалій, $\mathbf{X} \in \mathbb{R}^{t \times m}$ з кроком в часі $t \in \mathbb{N}$ та вимірами $m \in \mathbb{N}$. Початкові дані X та прогнозовані \hat{X} мають бути ідентичні, але в реальних задачах це рідко зустрічається. Тому необхідно визначити функцію втрат $Loss(X, \hat{X})$ для оцінки якості прогнозування. Для функції втрат зазвичай використовують MSE або MAE метрики. Детально про ці метрики зазначено в наступному розділі 2.

У випадку датасету який містить аномальні дані $\mathbf{A} \in \mathbb{R}^{t \times m}$ який поданий до моделі LSTM буде очікуватись наступний результат:

$$Loss(X, \hat{X}) < Loss(A, \hat{A}) \quad \forall x \in X, \hat{x} \in \hat{X}, a \in A, \hat{a} \in \hat{A} \quad (1.6)$$

де кожен зразок помічений як "нормальний" або "аномальний" в залежності від значення функції втрат. В реальних задачах для кількісної оцінки можливостей мережі використовується порогове значення:

$$c(A) = \begin{cases} \text{Normal Loss} (A, \hat{A}) < e_{th}, \\ \text{Anomaly Loss} (A, \hat{A}) \geq e_{th} \end{cases} \quad (1.7)$$

де e_{th} — заздалегідь визначене граничне значення. Оптимальний поріг вибрано належним чином, щоб максимізувати звичайні показники, а саме, Precision (P) і Recall (R) у випадку розміченого датасету [3]. Детально про ці метрики також зазначено в наступному розділі 2.

1.2 Сучасні нейромережеві моделі для виявлення аномалій на основі підходу реконструкції

Означення 1.5. Методи реконструкції для виявлення аномалій у часових рядах — це методи, що використовуються для ідентифікації та виявлення аномалій або незвичайних закономірностей у безперервних даних. Вони передбачають реконструкцію або відтворення вихідних даних часового ряду на основі моделі або представлення і порівняння їх зі спостережуваними або фактичними точками даних. Значні відмінності між реконструйованими та спостережуваними даними можуть вказувати на наявність аномалій.

Багато складних методів виявлення аномалій часових рядів ґрунтуються на моделюванні часових рядів для прогнозування майбутніх значень і помилок прогнозування. Попри це, не існує надійних моделей на основі прогнозування, які можуть створювати точні моделі для швидких і безперервно мінливих часових рядів, оскільки часові ряди можуть швидко змінюватися в будь-який момент часу [4]. Через те, що такі часові ряди неможливо передбачити заздалегідь, виявлення аномалій на основі прогнозування є неефективним. Моделі, що базуються на прогнозуванні,

мають різке збільшення помилки прогнозування зі збільшенням кількості часових рядів [5]. З цієї причини наявні моделі нездатні робити дуже короткострокові прогнози та виявляти наступні аномалії з прийнятною точністю.

Моделі на основі реконструкції будуються шляхом кодування підпослідовності нормальних навчальних даних у латентному просторі (низької розмірності). Вхідними даними для моделі є ковзне вікно, яке забезпечує часовий контекст для процесу реконструкції. Модель навчається лише на нормальніх даних, тому неможливо реконструювати аномальні підпослідовності під час фази тестування. В результаті, аномалії виявляються шляхом реконструкції точок/підвікон з тестових даних і порівняння їх з фактичними значеннями, що називається помилкою реконструкції. У деяких моделях аномальні точки/підпослідовності мають низьку ймовірність реконструкції, тому виявлення аномалій спрацьовує, коли ймовірність реконструкції падає нижче певного порогу.

Перед тим як більш детально розглянути модель, що буде реалізована в цій роботі варто ознайомитися з прикладом простої AE моделі з LSTM шарами на основі підходу реконструкції.

Означення 1.6 (англ. AutoEncoder, AE). Автокодувальник — це тип ANNs, призначений для неконтрольованого навчання та зменшення розмірності. Він спрямований на вивчення стисненого представлення вхідних даних, також відомого як прихованій простір.

AE модель (Рис. 1.2) складається з двох LSTM модулів: кодера і декодера.

Означення 1.7 (англ. Encoder). Енкодер — це компонент AE моделі який приймає вхідні дані та перетворює їх на інше представлення. Процес кодування зазвичай передбачає зменшення розмірності вхідних даних або вилучення їх найбільш відповідних характеристик. Енкодер відповідає за захоплення важливої інформації з вхідних даних і створення стиснутого

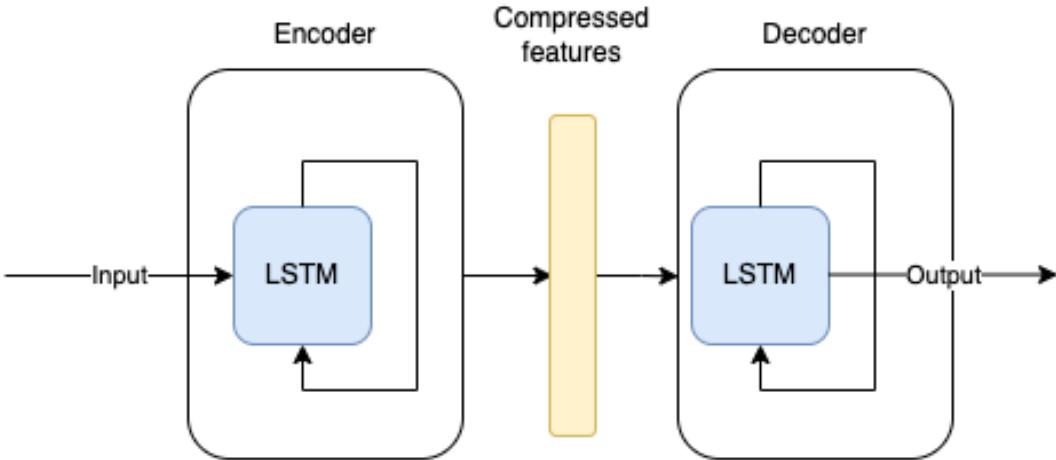


Рисунок 1.2 – LSTM Autoencoder модель

представлення.

Означення 1.8 (англ. Decoder). Декодер є аналогом енкодера. Він бере закодоване представлення, створене кодером, і реконструює або декодує його назад у форму, подібну або еквівалентну вихідним вхідним даним. Декодер має на меті повернути назад процес кодування, відновлюючи інформацію, втрачену під час кодування. Він відображає стиснуте представлення назад у вихідний простір даних.

Дані надходитимуть на енкодер, який вивчатиме базові особливості процесу. Ці функції в основному мають зменшений вимір. З іншого боку, декодер відтворить вихідні дані з цих основних функцій. На виході було б зменшено шум і розміри. По суті, АЕ може ефективно стискати функції з даних великої розмірності. Використання АЕ є дуже популярним у проблемах виявлення аномалій, оскільки АЕ може значно підвищити точність виявлення аномалій, яку інші методи, такі як PCA, не зможуть виконати [6]. Одною з особливостей АЕ моделей є латентний простір, що міститься між кодером та декодером.

Означення 1.9 (англ. Latent space). Латентний простір — це простір низької розмірності, де дані стискаються. Він може бути уявлений як простір, в якому подібні дані знаходяться близько одного, а

різні дані знаходяться далеко один від одного.

Тобто, вектор у латентному просторі представляє сутність даних із збереженою інформацією про них.

Ще одна перевага використання АЕ полягає в тому, що його легко навчити і не вимагає обчислювальної складності, з якою стикаються інші методи, такі як PCA ядра. Архітектура АЕ могла обробляти великомасштабні дані та проводити вибір функцій, що зробило її відповідним вибором для застосування в цій дослідницькій області [7, 8].

1.3 Нейронні мережі на основі реконструкції з використанням багатосигнальної синхронності

Означення 1.10 (англ. Multisignal synchrony). Багатосигнальна синхронність — це ситуація, коли два або більше сигнали або події відбуваються або спостерігаються одночасно або зі значним взаємозв'язком у часі. Це означає, що різні сигнали або події спостерігаються в одинаковий часовий момент або з деяким стабільною затримкою одне відносно іншого.

Багатосигнальна синхронність може мати місце в різних контекстах і дисциплінах. Наприклад, у мережах зв'язку багатосигнальна синхронність належати до ситуації, коли різні сигнали або пакети даних надсилаються та отримуються зі спільною точкою синхронізації. Це може бути досягнуто за допомогою спеціальних протоколів або алгоритмів, що забезпечують узгодженість часу між відправниками та отримувачами.

У сфері сенсорних даних, таких як біомедичні сигнали чи аналіз часових рядів, багатосигнальна синхронність може вказувати на наявність взаємозв'язків або кореляцій між різними сигналами. Наприклад, якщо два сигнали (наприклад, ЕЕГ та ЕКГ) мають подібні фазові залежності або співвідношення в часі, це може свідчити про багатосигнальну синхронність між ними, що може бути важливим для

аналізу та вивчення взаємодії між різними системами або фізіологічними процесами.

В загальному розумінні багатосигнальна синхронність вказує на співвідношення часу або взаємозв'язок між різними сигналами або подіями та може мати важливе значення в багатьох областях, де вивчається та аналізується часова структура даних

Оцінка багатосигнальної синхронності добре розглядається в літературі з обробки сигналів. Однак, якщо мова йде про виявлення аномалій у реальному часі то вплив асинхронності на навчання та продуктивність не враховується, хоча це стан більшості реальних багатовимірних часових рядів. Тому інженерами компанії eBay для розв'язання задачі пошуку аномалій в багатовимірних асинхронних часових рядах була розроблена модель RANSynCoders [9]. Архітектура моделі зображена на Рис. 1.3.

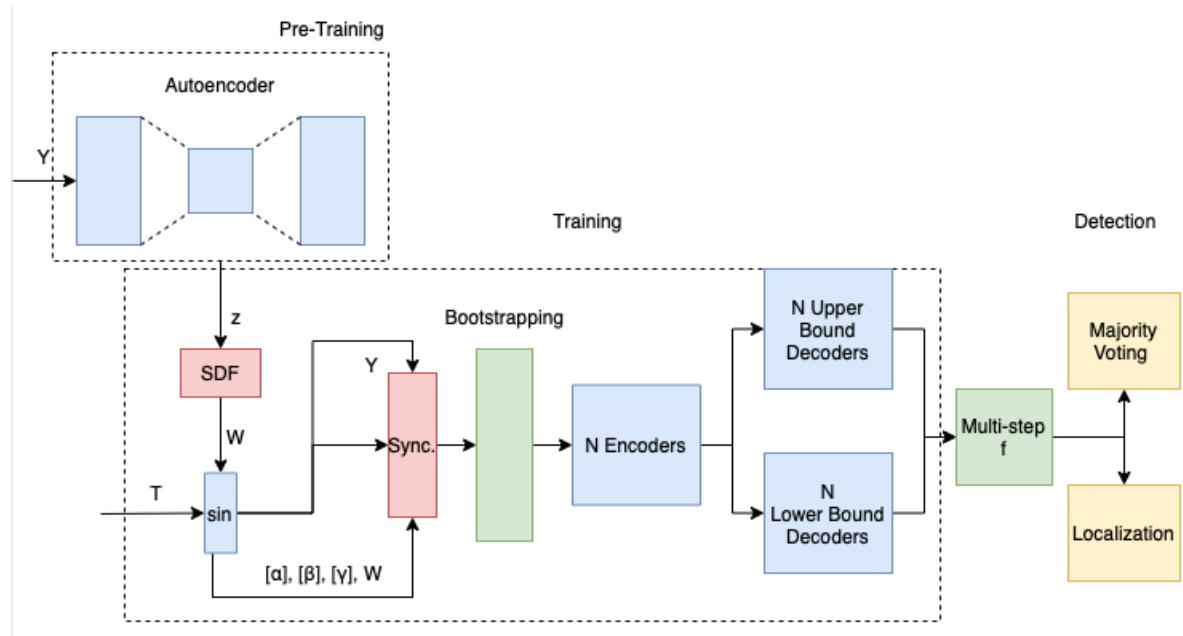


Рисунок 1.3 – AE RANSynCoders

В даній моделі першим етапом є навчання АЕ підмоделі, використовуючи екстремальний латентний простір розміром 1. Для мети навчання необхідно мінімізувати втрати 50-го квантиля реконструкції, що

позначається як $\min L_q = 0,5(Y, \bar{Y})$. Втрата 50-го квантиля еквівалентна медіані залишків, що є придатним показником для роботи з викидами. Потім використовується спектральний аналіз зі швидким перетворенням Фур'є на одновимірному прихованому представленні, щоб визначити вектор частоти W розміром S . Вектор W містить частоти, які теоретично вважаються частотами, що домінують та зазвичай присутніми в характеристиках серії, де S є кількістю синусоїдальних компонентів, достатніх для апроксимації багатовимірного часового ряду Y .

Далі необхідно змоделювати асинхронний багатовимірний часовий ряд. Він моделюється за наступною формулою:

$$y_{i,t_i} = \left[\sum_{s=0}^S \alpha_{s,i} \sin(\omega_s(t_i + \beta_{s,i})) \right] + \gamma_i + \epsilon_{i,t_i} \quad (1.8)$$

де t_i – час вибірки або епохи (може бути непослідовним для різних сигналів), S – це синусоїдна компонента, $\alpha_{s,i}, \beta_{s,i}, \gamma_i$ – ваги навчання сигналу, ω_s – загальні частоти що домінують, ϵ_{i,t_i} – помилка навчання.

Використовуючи градієнтний спуск, ω і β неможливо ефективно вивчити через нелінійність. Оскільки нелінійні функції можуть мати багато локальних мінімумів, оптимізація може стати важкою задачею. При застосуванні градієнтного спуску до нелінійних функцій, можуть виникати проблеми, такі як в'язкість градієнта, затухання або зростання градієнта. Ці проблеми можуть ускладнювати процес навчання та зменшувати його ефективність. Тому тут застосовується наближення шляхом лінеаризації кутової частини рівняння як $\omega_s t_i + \hat{\beta}_{s,i}$, де $\beta = \frac{\hat{\beta}}{\omega}$.

Член шуму ϵ обчислюється шляхом віднімання необробленого сигналу від підігнаних синусоїдальних компонентів. Відповідно, ваги для α , β і γ оптимізуються під час зворотного поширення.

Залишилось лише виразити шум, як показано далі:

$$\epsilon_{i,t_i} = \alpha_{\epsilon,i,t_i} \sin(\omega_0(t_i + \beta_{\epsilon,i,t_i})) \quad (1.9)$$

В даному рівнянні β_{ϵ,i,t_i} виражається наступним чином:

$$\beta_{\epsilon,i,t_i} = -t_i + \frac{v_0}{4} \quad (1.10)$$

де $v_0 = \frac{2\pi}{\omega_0}$ – частота на часовому кроці. Виходячи з цього отримуємо:

$$\epsilon_{i,t_i} \approx \alpha_{\epsilon,i,t_i} \quad (1.11)$$

Використовуючи формули, представлені вище, можна вирівняти часові ряди до системи відліку \bar{t} :

$$y_{i,\bar{t}_i} = \left[\sum_{S=0}^S \alpha_{s,i} \sin(\omega_s (t_i + \beta_{s,i} - \bar{\beta}_i)) \right] + \gamma_i + \epsilon_{i,\bar{t}_i} \quad (1.12)$$

де \bar{t}_i – опорний кадр (синхронізований час вибірки), $\bar{\beta}_i$ – оптимізований фазовий зсув, тобто:

$$\bar{\beta}_i = \beta_{0,i} + (t_i - \bar{t}) \quad (1.13)$$

ϵ_{i,\bar{t}_i} – проекція помилки на \bar{t} , тобто:

$$\epsilon_{i,t_i} = \epsilon_{i,\bar{t}_i} \sin\left(\omega_0 \left(\frac{v_0}{4} - \bar{\beta}_i\right)\right) \quad (1.14)$$

Синхронізація призводить до збільшення мультиколінеарності, таким чином зменшуючи кількість інформації та розмірність латентного простору. Приклад цього продемонстровано на Рис. 1.4:

Після того як ряди стали синхронізовані випадковим чином обирається певна кількість ознак ряду як вхідні дані до кожного кодеру [10]. Далі декодери реконструюють верхні та нижні границі. Це буде більш ефективно ніж реконструювати сам сигнал, через випадкову поведінку природних даних. Тому декодери намагаються мінімізувати втрати квантильної реконструкції, яка виражена наступним чином:

$$L_{q=\delta} = \sum_{i=0}^I \max \left[\delta \left(\hat{y}_{i,\bar{t},n}^{LB} - y_{i,\bar{t}} \right), (\delta - 1) \left(\hat{y}_{i,\bar{t},n}^{LB} - y_{i,\bar{t}} \right) \right], \forall n \in N \quad (1.15)$$

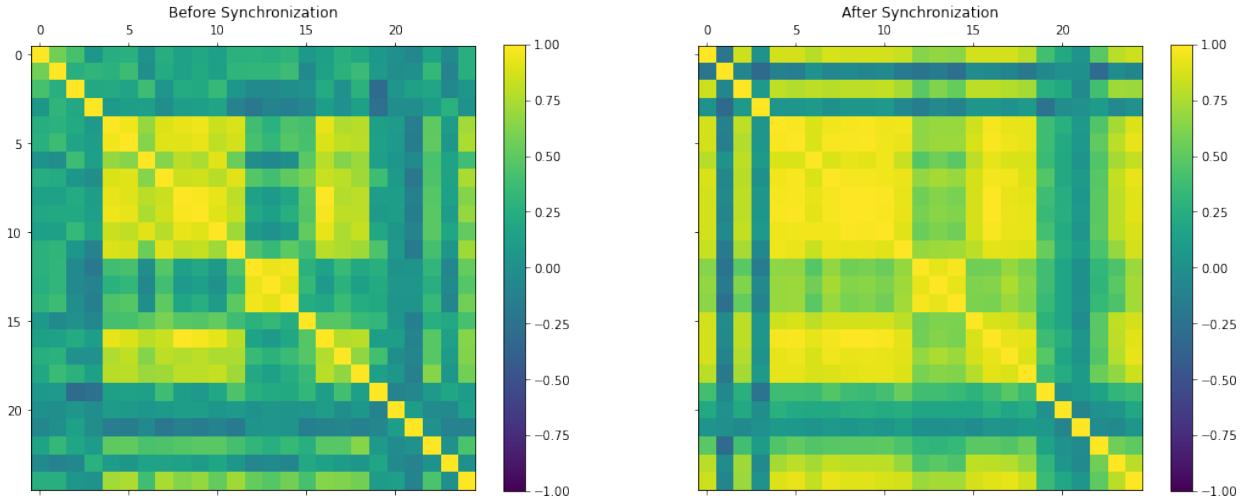


Рисунок 1.4 – Збільшення мультиколінеарності після синхронізації

де δ — параметр квантиля, який довільно вибирається як половина переконання користувача щодо кількості шуму або аномалій, присутніх у даних. Analogічна формула отримується і для верхньої межі:

$$L_{q=\delta} = \sum_{i=0}^I \max \left[\delta \left(\hat{y}_{i,\bar{t},n}^{UB} - y_{i,\bar{t}} \right), (1 - \delta) \left(\hat{y}_{i,\bar{t},n}^{LB} - y_{i,\bar{t}} \right) \right], \forall n \in N \quad (1.16)$$

В ідеалі кількість прихованих шарів декодера має бути принаймні на один шар більшою, ніж приховані шари кодерів, оскільки вихідний розмір декодера набагато більший, ніж вхід кодерів. Завдяки одночасному навчанню AE_{UB} і AE_{LB} з протилежними функціями втратами спостерігаємо, що вагові коефіцієнти кодерів збігаються швидше й ефективніше, ніж інші методи, без необхідності регуляризації чи інших методів пом'якшення викидів. Для більшості наборів даних знадобилося менш ніж десять епох.

Останнім кроком залишилось визначення аномалій. Для цього є формул, яка виглядає так:

$$P(i_n) = \begin{cases} 0 & \text{if } \hat{y}_{i,\bar{t},n}^{LB} < y_{i,\bar{t}} < \hat{y}_{i,\bar{t},n}^{UB} \\ 1 & \text{if otherwise} \end{cases}, \forall i \in I, n \in N \quad (1.17)$$

Як результат буде отримана матриця розміру $I \times N$ з бінарними значеннями. Таким чином, висновок про аномалії шляхом робиться шляхом голосування більшістю, де спостереження позначається як аномальне, якщо на виході понад 1, ніж 0. Цей підхід є більш практичним для промисловості, ніж підхід апостеріорного налаштування порогу.

Висновки до розділу 1

У першому розділі роботи було розглянуто огляд нейромережевих методів, що використовуються для виявлення аномалій. Було розглянуто підхід на основі прогнозування, де навчена модель використовується для прогнозування точки або послідовності на основі вхідних даних. За допомогою порівняння прогнозованих значень з фактичними значеннями можна визначити аномалії. Було також представлено приклад моделі з підходом на основі прогнозування, в якому використовуються штучні нейронні мережі (ANNs), рекурентні нейронні мережі (RNN) та довга короткочасна пам'ять (LSTM) для аналізу багатовимірних часових даних.

Також були розглянуті АЕ моделі, що є простими у навченні та не вимагають складних обчислювальних операцій, що робить їх популярними у виявленні аномалій. Їх легко застосовувати до великих даних і вони здатні здійснювати вибір функцій для ефективного аналізу даних.

Як удосконалення АЕ була розглянута модель RANSynCoders, яка базується на концепції багатосигнальної синхронності. Модель використовує АЕ для реконструкції та синхронізації рядів. Завдяки одночасному навчанню АЕ з протилежними функціями втрат, вагові коефіцієнти кодерів збігаються швидше й ефективніше, що сприяє ефективній роботі моделі без додаткової регуляризації чи методів пом'якшення викидів. Висновок про аномалії здійснюється шляхом голосування більшістю, де спостереження позначається як аномальне, якщо на виході більше ніж 1, ніж 0. Цей підхід є практичним для

промислових застосувань.

Як результат проведеного огляду моделей було отримане теоретичне підґрунтя про фундаментальні підходи до розв'язання задачі виявлення аномалій в багатовимірних часових рядах. Також варто зазначити, що у цій галузі активно розвивається та інженери великих компаній, такі як eBay зацікавлені в удосконаленні методів що уже існують.

Отже, можна зазначити, що LSTM добре справляється з аналізом даних, які мають послідовний контекст, наприклад, часові ряди, де кожне значення залежить від попередніх значень. LSTM має внутрішні механізми пам'яті, що дозволяють йому зберігати і використовувати інформацію з попередніх часових кроків, навіть на великих відрізках часу. Тому він ефективно працює з даними, де довгострокові залежності грають важливу роль. LSTM може моделювати складні залежності в часовихрядах, такі як сезонність, тренди, циклічність та інші нелінійні шаблони.

На відміну від LSTM, AE слугує для витягування головних ознак з великого обсягу даних та зменшення їх розмірності. Він може бути корисним при аналізі даних з багатьма ознаками, де бажано знизити вимоги до обчислювальних ресурсів. AE може використовуватися для генерації нових прикладів даних, піддаючи шумові вхідні дані та отримуючи реконструйовані виходи. Це може бути корисним при синтезуванні нових прикладів для тренувального набору даних.

2 ОГЛЯД АЛГОРИТМІВ ПОШУКУ ПОРОГОВОГО ЗНАЧЕННЯ ТА МЕТРИКИ ОЦІНКИ ЯКОСТІ МОДЕЛЕЙ

У другому розділі даної роботи буде розглянуто метрики для оцінки якості як якості прогнозування часового ряду окремо для моделі RNN LSTM так і метрики для оцінки якості класифікації нормальних даних та аномальних. Це необхідно надалі для порівняння моделей та алгоритми для пошуку оптимального динамічного порогового значення (threshold).

2.1 Метрики якості моделей

Оцінка ефективності моделей є критичним етапом в будь-якому процесі машинного навчання і класифікації даних. Відповідно, для отримання надійний необхідно обрати правильні метрики.

Метрики FN, FP, TP, TN є ключовими метриками, які у задачах класифікації з метою оцінки продуктивності класифікаційних моделей. Вони є чотири основні категорії результатів, одержуваних при класифікації об'єктів [11].

False Negative (FN) - хибнонегативне рішення: FN є ситуацією, коли модель неправильно класифікує об'єкт як негативний (негативний), тоді як насправді він є позитивним (належить до позитивного класу).

$$FN = \sum_{i=0}^N 1(y_i = 1, \hat{y}_i = 0) \quad (2.1)$$

False Positive (FP) - хибнопозитивне рішення: FP виникає, коли модель неправильно класифікує об'єкт як позитивний, тоді як він насправді належить до негативного класу.

$$FP = \sum_{i=0}^N 1(y_i = 0, \hat{y}_i = 1) \quad (2.2)$$

True Positive (TP) - правильне позитивне рішення: ТР виникає, коли модель правильно класифікує об'єкт як позитивний, і він насправді належить до позитивного класу.

$$FP = \sum_{i=0}^N 1(y_i = 1, \hat{y}_i = 1) \quad (2.3)$$

True Negative (TN) - правильне негативне рішення: TN виникає коли модель правильно класифікує об'єкт як негативний, і він дійсно належати до негативного класу.

$$FP = \sum_{i=0}^N 1(y_i = 0, \hat{y}_i = 0) \quad (2.4)$$

Ці метрики часто використовуються разом або у поєднанні з іншими метриками, такими як точність, повнота та F1 міра, для повної оцінки якості класифікаційних моделей [12, 13]. Правильне розуміння та інтерпретація цих метрик є важливим кроком для ефективної оцінки та покращення продуктивності моделей класифікації.

F1-оцінка (F1 score), повнота (Recall) і точність (precision) є важливими метриками для оцінки ефективності моделей в задачах класифікації. У цьому невеликому рефераті ми розглянемо ці метрики, їх значення і використання при оцінці моделей.

F1-оцінка є гармонічним середнім між повнотою і точністю. Ця метрика використовується для вимірювання збалансованості між цими двома метриками. F1-оцінка може бути обчислена за допомогою наступної формулі [14]:

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (2.5)$$

де точність (precision) визначається як відношення ТР (True Positive) до ТР та FP (False Positive), а повнота (Recall) - як відношення ТР до ТР та FN (False Negative).

Повнота (Recall) вимірює відсоток правильно визначених позитивних прикладів серед усіх дійсних позитивних прикладів. Вона обчислюється за допомогою наступної формули:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.6)$$

Точність (precision) вимірює відсоток правильно класифікованих позитивних прикладів серед усіх прикладів, які були визначені як позитивні. Вона визначається за формулою:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.7)$$

Ці метрики допомагають отримати повну картину про ефективність моделей. Повнота важлива в задачах, де важливо максимізувати виявлення позитивних прикладів.

Оцінка моделі за допомогою F1-оцінки, Recall і Precision дозволяє врахувати як правильно класифіковані позитивні приклади, так і наявність помилкових класифікацій. Це допомагає отримати більш об'єктивну оцінку ефективності моделі.

Також для оцінки моделі на основі прогнозування необхідно описати метрики. В цій роботі були обрані метрики MSE та MAE. Вони обраховуються як показано нижче.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.8)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.9)$$

де y_i — реальне значення функції, \hat{y}_i — передбачене значення функції.

MAE вимірює середню абсолютну різницю між прогнозованими значеннями і фактичними значеннями. Це означає, що для кожного прогнозу і його відповідного фактичного значення, віднімається одне від іншого, результат береться за модулем (абсолютне значення) і додається

до загальної суми помилок. МАЕ не зважує на розмір різниці, а просто обчислює середнє значення абсолютних помилок. Це робить МАЕ менш чутливим до великих відхилень.

MSE вимірює середню квадратичну різницю між прогнозованими значеннями і фактичними значеннями. Воно обчислює квадрат кожної різниці, додає всі квадрати разом і обчислює середнє значення. MSE квадратично збільшує значення великих помилок, що робить його більш чутливим до великих відхилень.

Отже, МАЕ та MSE використовуються для вимірювання помилок у прогнозуванні, але МАЕ надає інформацію про середню абсолютну різницю, тоді як MSE надає інформацію про середньоквадратичну різницю.

2.2 Алгоритм сліпого пошуку порогового значення

Щоб підкреслити функціональність системи мажоритарного голосування в архітектурах, які розглядаються в цій роботі над пороговими підходами вибору, обчислюється F1-оцінки та інших метрик (precision, recall, FPR, TP, TN, FP, FN) для послідовності оцінок (score) та відповідних міток належності до класу аномальних або нормальніх даних при використанні певного порогового значення (Далі в роботі це буде позначатися як calc_seq). Додатково обчислюються ці самі оцінки, але які були відновлені на основі нерухомих вікон часу (non-rolling time-window), тобто повторно дискретизовану в не проміжному часовому вікні (Далі в роботі це буде позначатися як calc_twseq).

Algorithm 2.1 Алгоритм сліпого пошуку найкращого порогового значення на основі F1 оцінки

```

blind_bf_search(score, label, val, start, end, step_num, tw, normal)

Results and labels score_val, label_val = val

Guess g =  $\frac{start+end}{2}$ 

Search range range = end - start

Threshold th = start

F1, Precision, Recall m = (-1., -1., -1.)

Result threshold m_t = 0

for i = 0 to step_num do

    Increase threshold th +=  $\frac{range}{search\_num}$ 

    Make prediction pred = score_val < th

    if max(|label_val - normal|) or max(pred) then

        Metrics result target = calc_twseq(score_val, label_val, normal, th, tw, pred)

        if target[0] > m[0] then

            m = target

            m_t = th

        else

            Continue

        end if

    end if

end for

The best threshold th = m_t

The best metrics m = calc_twseq(score_val, label_val, normal, th, tw, pred)

Return th, m

```

Запропонований алгоритм 2.1 сліпого пошуку порогового значення є імітацією випадку коригування порогу після розгортання моделі на практиці [15]. Суть алгоритму полягає в тому, що необхідно поділити набір даних на якому буде перевірятися робота на дві половини та визначати порогове значення на основі спостережуваної першої половини, а потім застосовуємо його до другої половини. Таким чином, ми можемо використати запропоноване точкове коригування з урахуванням затримки та розрідженності для цієї стратегії за замовчуванням.

Алгоритм blind_bf_search 2.1 реалізує пошук найкращого значення

порогу для досягнення максимального значення F1-міри в задачі виявлення аномалій.

Алгоритм отримує на вхід оцінки аномалій (score), цільові мітки (label), кортеж або список результатів і міток, які будуть використовуватися для налаштування порогу (val), тобто друга половина даних, початкове значення порогу (start), кінцеве значення порогу (end), кількість кроків для пошуку від початку до кінця (step_num), частота повторної дискретизації для уникнення перерахунку ТР і FP або недорахування FN і TN (tw), значення нормальної поведінки (normal).

Цей алгоритм служить для автоматичного знаходження оптимального порогового значення, яке допомагає виявити аномалії або позитивні випадки в даних на основі F1 метрики, яка компромісно поєднує точність і повноту.

Основні кроки алгоритму:

1) Алгоритм починає з розрахунку середнього значення порогу (Guess) шляхом обчислення середнього значення початкового і кінцевого значення.

2) Далі визначається діапазон пошуку (Search range), який обчислюється як різниця між початковим і кінцевим значеннями.

3) Встановлюється початкове значення порогу (Threshold).

4) Ініціалізуються змінні F1, Precision, Recall (m) зі значеннями -1.

5) Змінна Result threshold (m_t) ініціалізується значенням 0.

6) Починається цикл з кроками від 0 до step_num.

7) Виконується прогнозування (Make prediction) на основі поточного порогового значення, отримуючи змінну pred.

8) Перевіряється, чи є максимальне значення різниці між мітками й нормальним значенням або максимальне значення pred (чи є хоча б одне значення True).

9) Якщо умова виконується, виконується обчислення метрик результату (Metrics result) шляхом виклику функції calc_twseq.

10) Якщо значення F1 метрики для поточного порогу більше, ніж

значення $m[0]$ (попередньо найкращої $F1$ метрики), оновлюється змінна m (метрики) і m_t (порогове значення, для якого досягнута найкраща метрика).

11) Якщо значення $F1$ метрики не є кращим, алгоритм продовжує ітерацію циклу.

12) Після закінчення циклу, найкраще порогове значення зберігається в змінній th , а найкращі метрики обчислюються за допомогою функції $calc_twseq$ з найкращим th .

13) На виході алгоритм повертає значення th (найкраще порогове значення) і m (найкращі метрики).

2.3 Алгоритм пошуку найкращого теоретичного порогового значення

Найкращий $F1$ ($F1best$) є алгоритмом що реалізує оцінки суто теоретично, тобто досягти цієї оцінки можливо, але це складно реалізувати в реальних задачах, оскільки алгоритм базується на виборі оптимального порогу шляхом випробування всіх можливих порогів для результатів виявлення (Brute-force) [16]. Хоча може бути так, що одна модель є кращою, якщо відомий оптимальний поріг, неможливо досягти цього на практиці без вичерпного періоду маркування аномалій після розгортання та зворотного зв'язку для кожного випадку використання. Ми за замовчуванням застосовуємо метод точкового коригування в цій стратегії. Псевдокод алгоритму 2.2 наведений нижче.

Algorithm 2.2 Алгоритм пошуку найкращого теоретичного порогового значення

```

best_bf_search(score, label, start, end, step_num)

Search range range = end - start

Threshold th = start

F1, Precision, Recall m = (-1., -1., -1.)

Result threshold m_t = 0

for i = 0 to step_num do

    Increase threshold th +=  $\frac{\text{range}}{\text{search\_num}}$ 

    Make prediction pred = score > th

    Metrics result target = calc_seq(score, label, th, pred)

    if target[0] > m[0] then

        m = target

        m_t = th

    end if

end for

Return m_t, m

```

Алгоритм best_bf_search реалізує пошук найкращого значення порогу для досягнення максимального значення F1-міри в задачі виявлення аномалій. Алгоритм отримує на вхід оцінки аномалій (score), мітки (label), початкове значення порогу (start), кінцеве значення порогу (end) і кількість кроків (step_num). На відміну від сліпого пошуку в цьому алгоритмі не треба ділити дані на дві частини.

Цей алгоритм працює наступним чином:

- 1) Ініціалізується змінна m_t для збереження найкращого порогового значення, а також змінна m для збереження найкращої F1-оценки та інших метрик (precision, recall).
- 2) Запускається цикл for змінною i від 0 до step_num.
- 3) Збільшується значення порогу th на кожній ітерації циклу, розподіляючи діапазон між start та end на рівні частини.
- 4) Виконується прогнозування pred, де елементи послідовності score порівнюються з пороговим значенням th. Якщо елемент score більше за th, відповідний елемент pred встановлюється на True, інакше - на False.

5) Викликається функція calc_seq, яка обчислює F1-оцінку та інші метрики для послідовності score та відповідних міток label з використанням порогового значення th та прогнозів pred. Отримані метрики зберігаються в змінній target.

6) Порівнюється значення F1-оцінки target[0] з найкращою F1-оцінкою. Якщо target[0] більше, ніж m[0], то оновлюється змінна m з отриманими метриками та змінна m_t з поточним значенням порогу th.

7) Після завершення циклу, найкраще порогове значення m_t та найкращі метрики m повертаються з алгоритму.

Отже, алгоритм best_bf_search послідовно перебирає різні значення порогу в заданому діапазоні та обчислює F1-оцінку для кожного значення. Найкраще порогове значення визначається як те, яке має найвищу F1-оцінку.

Висновки до розділу 2

В даному розділі було розглянуто метрики для оцінки якості моделей, як для прогнозування часових рядів у моделі RNN LSTM, так і для класифікації нормальніх та аномальних даних. Метрики FP, FN, TP, TN є ключовими метриками для оцінки продуктивності класифікаційних моделей. Додатково, для оцінки якості класифікаційних моделей також використовуються метрики точності (precision), повноти (recall) і F1-оцінки. F1-оцінка є гармонічним середнім між точністю і повнотою, і використовується для вимірювання збалансованості між цими двома метриками. Точність визначає відсоток правильно класифікованих позитивних прикладів серед усіх прикладів, які були визначені як позитивні. Полнота вимірює відсоток правильно визначених позитивних прикладів серед усіх дійсних позитивних прикладів. Крім того, для оцінки моделей з використанням прогнозування були обрані метрики середньої квадратичної помилки (MSE) і середньої абсолютної помилки (MAE).

Наступним кроком в цьому розділі було запропоновано алгоритм сліпого пошуку порогового значення для досягнення максимального значення F1-оцінки в задачі виявлення аномалій. Алгоритм використовує послідовну стратегію налаштування порогу, де перша половина даних використовується для визначення порогу, а друга половина даних використовується для оцінки виявлення аномалій з використанням цього порогу.

Алгоритм F1best використовує метод brute-force, що означає перебір усіх можливих порогів для виявлення аномалій з метою вибору оптимального порогу. У процесі роботи алгоритм змінює порогове значення і обчислює метрики F1, точність та повноту для кожного порогу. Завдяки цьому алгоритм може визначити оптимальне порогове значення, яке максимізує F1-міру.

Як результат проведених досліджень, були отримані методології оцінювання, що в повному обсязі зможуть дати представлення про ефективність моделей.

Отже, метрики є важливою складовою для оцінки ефективності будь-якої моделі. Правильний вибір метрик та методології оцінювання не менш важливе чим побудова самих моделей. Тому варто детально підходити до проблеми вибору метрик, щоб отримати вичерпну відповідь про ефективність моделей.

3 РЕАЛІЗАЦІЯ ТА АНАЛІЗ МОДЕЛЕЙ

У даному розділі буде застосовано розглянуті у попередніх частинах моделі, алгоритми та метрики для задачі виявлення аномалій в багатовимірних даних на практиці. А саме, спочатку будуть розглянуті датасети, на яких моделі навчалися. Буде розглянуто попередню обробку даних. Також більш детально звернемо увагу на те як будується архітектура, які гіперпараметри та параметри будуть використані. В кінці порівняємо результати моделей, та виявимо сильні та слабкі сторони моделей.

3.1 Підготовка даних

Для виконання цієї роботи були використані наступні пакети програмного забезпечення:

- python 3.9.6
- pandas 1.5.2
- tensorflow 2.11.0 (cpu build)
- keras 2.11.0 (cpu build)
- numpy 1.24.1
- scipy 1.9.3
- sklearn 1.2.0

Для порівняння моделей глибинного навчання на основі RNN LSTM та AE RanSynCoders моделі були обрані 2 публічних датасета, які відповідають необхідним критеріям для виконання роботи. А саме: датасети повинні містити багато ознак, тобто бути багатовимірними, датасет повинен складатися з двох частин: частина де дані, не містять аномалій та з частини, де вже наявні аномалії, а також, що не менш важливе другі частини кожного датасету повинні бути розмічені, тобто

повинен міститися файл з позначеними точками, які належать до нормальних даних або до аномальних, щоб якісно оцінити результат виявлення аномалій.

Тому для цієї роботи були обрані наступні датасети: SMD (Server Machine Dataset): набір даних SMD складається з даних, зібраних протягом 5 тижнів із 38 параметрами. Перша частина датасету містить 28479 записів, друга — 28479 і відповідну кількість міток класів; PSM (Pooled Server Metrics) — набір даних PSM запропонований eBay і складається з 26-вимірних даних, отриманих внутрішньо з вузлів сервера додатків. Перша частина датасету містить 132481 запис, друга — 87841 і відповідну кількість міток класів.

Попередня обробка датасету є важливим кроком у підготовці даних для подальшого аналізу та моделювання. Вона охоплює ряд операцій, спрямованих на покращення якості даних і забезпечення їх придатності для використання. Основні дії які були зроблені для обробки даних — заповнення пропущених даних середнім значенням, та нормалізація даних. Цей крок включає приведення даних до одного масштабу. Нормалізація може бути виконана шляхом перетворення значень в межах від 0 до 1 або -1 до 1. Масштабування допомагає уникнути впливу великих числових значень на аналіз. В даній роботі використовувався MinMaxScaler для нормалізації даних в межах від 0 до 1 за наступною формулою:

$$\text{MinMaxScaler: } x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3.1)$$

де x — оригінальне значення, x_{\min} — мінімальне значення у наборі даних, x_{\max} — максимальне значення у наборі даних, x_{scaled} — масштабоване значення.

Далі, щоб підготувати дані для тренування моделі необхідно використати TimeseriesGenerator. TimeseriesGenerator - це клас, який надає можливість створювати часові послідовності для задач прогнозування за допомогою методу генерації прикладів. Він забезпечує

автоматичне створення пакетів даних для навчання моделі з використанням часових послідовностей.

Основна ідея TimeseriesGenerator полягає в тому, щоб використовувати певну кількість попередніх відліків (залежить від визначеного вікна) для прогнозування майбутніх значень. Він приймає на вхід послідовність даних (наприклад, часовий ряд) і генерує пари (X, y) , де X - послідовність відліків, а y - майбутнє значення, яке потрібно прогнозувати.

Цей підхід дозволяє ефективно використовувати часові дані для прогнозування. TimeseriesGenerator спрощує процес підготовки даних і дозволяє швидко створювати пакети для навчання моделей прогнозування часових послідовностей.

3.2 Реалізація RNN LSTM та SynLSTM моделей

Для обох датасетів використовувалась одна модель. Модель складається з двох шарів LSTM, за якими йдуть активація LeakyReLU. LeakyReLU (Leaky Rectified Linear Unit) — це функція активації, яка зазвичай використовується в моделях глибокого навчання. Це різновид функції активації Rectified Linear Unit (ReLU). Функція LeakyReLU вводить невеликий нахил до негативної частини функції, дозволяючи невеликий градієнт для негативних вхідних даних. Формула для функції активації LeakyReLU визначається так:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases} \quad (3.2)$$

Далі йде Dropout (метод "випадкового відкидання" нейронів під час тренування моделі) у 10% та повнозв'язний шар із лінійною функцією активації. Модель скомпільовано із втратою середньоквадратичної помилки (MSE) і оптимізатором Адама з параметром 0.001. LSTM_1 та

LSTM_2 використовує активаційну функцію $tanh$. Кількість епох навчання 100, розмір батчу 32. Архітектура LSTM моделі показана в таблиці 3.1.

Layer (type)	Output Shape	Param
LSTM_1 (LSTM)	(None, 64, 32)	12416
LSTM_2 (LSTM)	(None, 32)	8320
LeakyReLU (LeakyReLU)	(None, 32)	0
Dropout (Dropout)	(None, 32)	0
Dense (Dense)	(None, 64)	2112
Total params: 22,848		
Trainable params: 22,848		
Non-trainable params: 0		

Таблиця 3.1 – Приклад архітектури RNN LSTM моделі

Для SynLSTM моделі архітектура залишається та сама, лише додається шар, що відповідає за синхронізацію часових рядів. В Таблиці 3.2 представлені метрики того як добре модель прогнозує часові ряди.

На наступних графіках можна побачити те, як якісно RNN LSTM модель здатна прогнозувати часові ряди. На Рис. 3.1 зображено результат прогнозування для перших 4 ознак PSM даних.

Model	Dataset	MAE	MSE
RNN LSTM	SMD	0.0024	0.0318
RNN SynLSTM	SMD	0.0004	0.0112
RNN LSTM	PSM	0.0077	0.0659
RNN SynLSTM	PSM	0.0027	0.0270

Таблиця 3.2 – Оцінка якості прогнозування часових рядів для RNN LSTM моделі

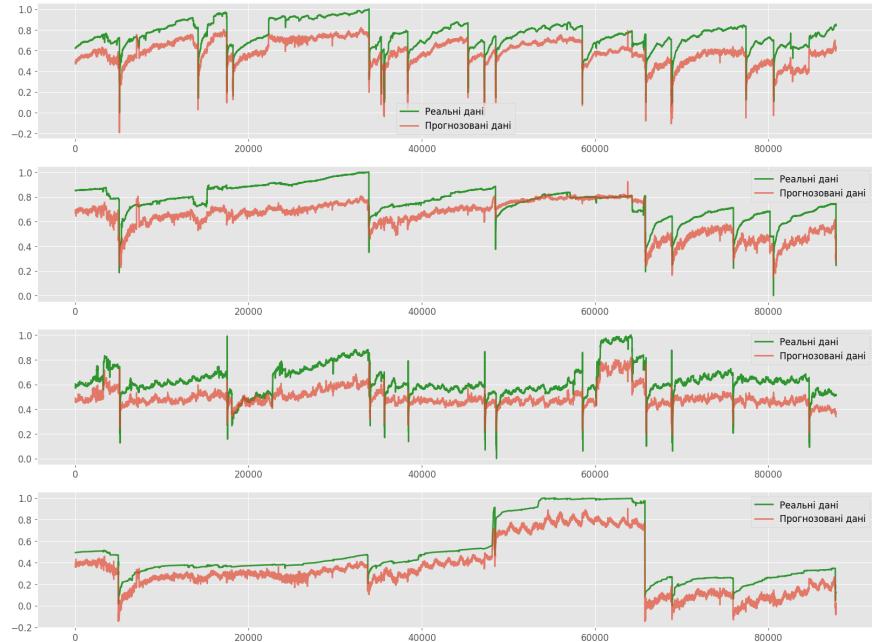


Рисунок 3.1 – Результат прогнозування часового ряду RNN LSTM
Моделлю для PSM набору даних

Аналогічний результат для перших 4 ознак даних SMD знаходиться на Рис. 3.2.

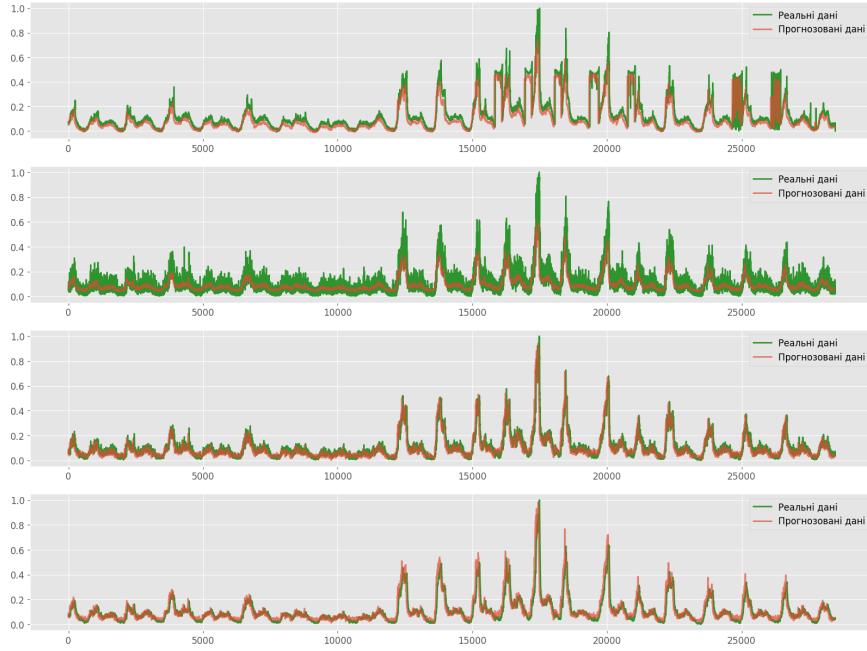


Рисунок 3.2 – Результат прогнозування часових рядів RNN LSTM Моделлю для SMD набору даних

Аналогічні результати, але для RNN SynLSTM моделі. на Рис. 3.3 та 3.4 чітко видно те, як видозмінилися часові ряди після синхронізації.

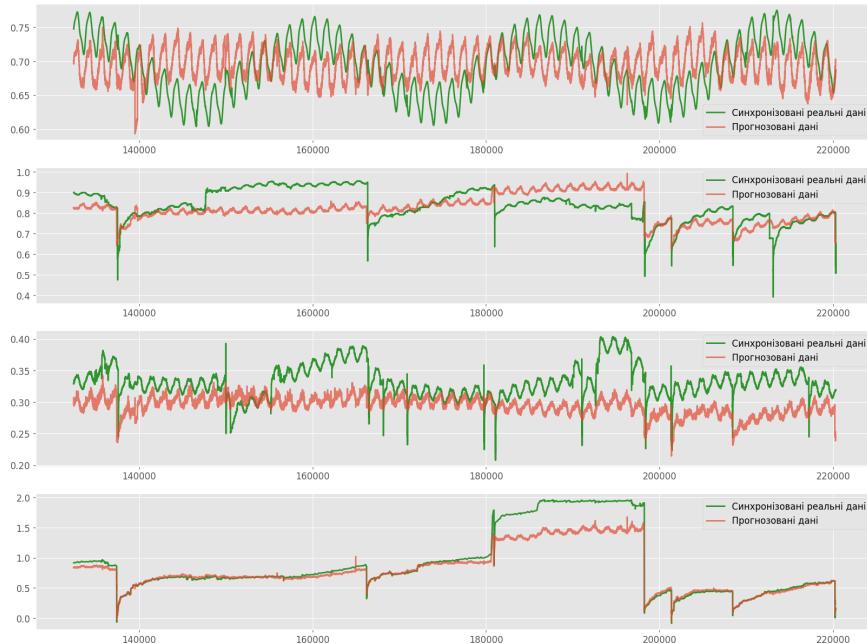


Рисунок 3.3 – Результат прогнозування часового ряду RNN SynLSTM Моделлю для PSM набору даних

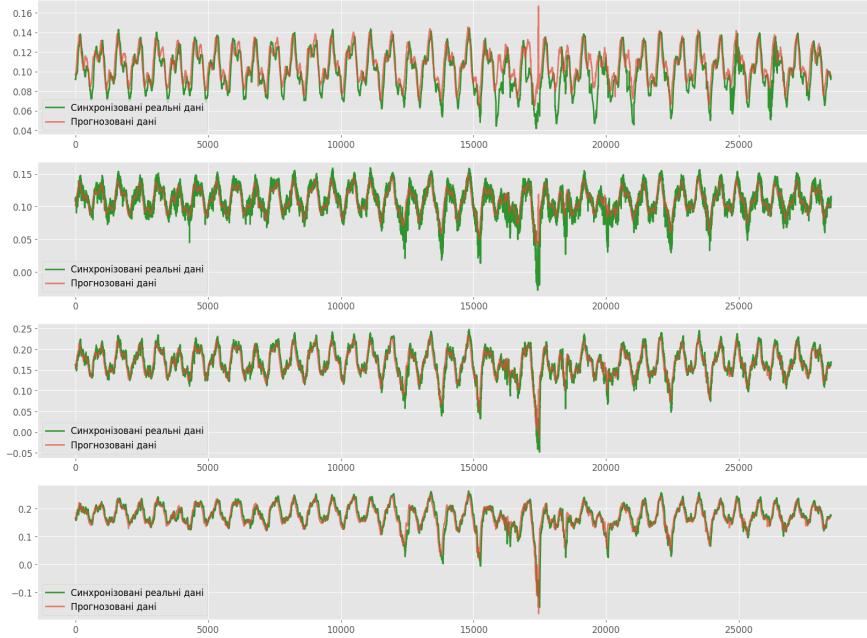


Рисунок 3.4 – Результат прогнозування часових рядів RNN SynLSTM
Моделлю для SMD набору даних

Результат для даних PSM знаходиться на Рис. 3.5.

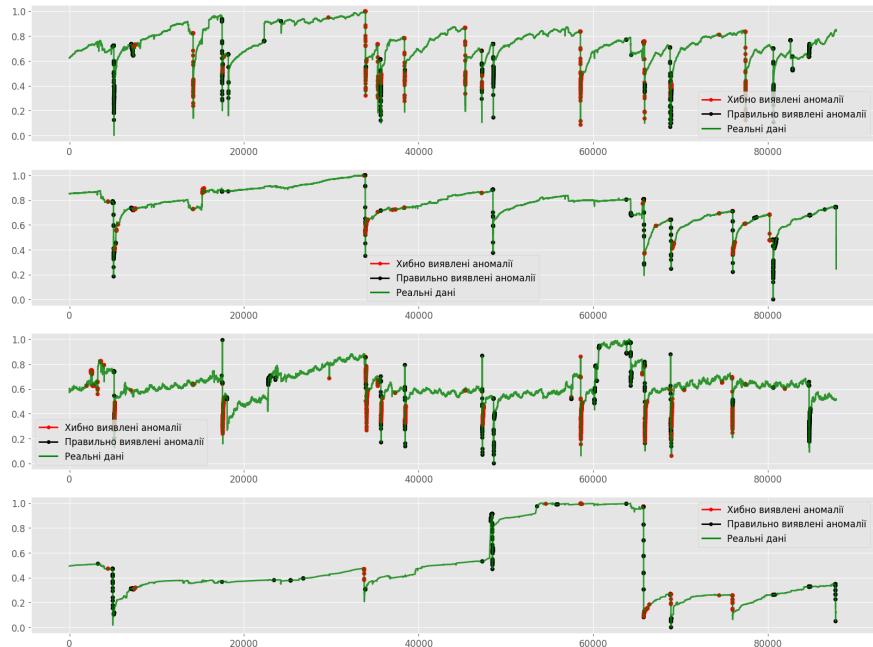


Рисунок 3.5 – Результат виявлення аномалій RNN LSTM Моделлю для
PSM набору даних

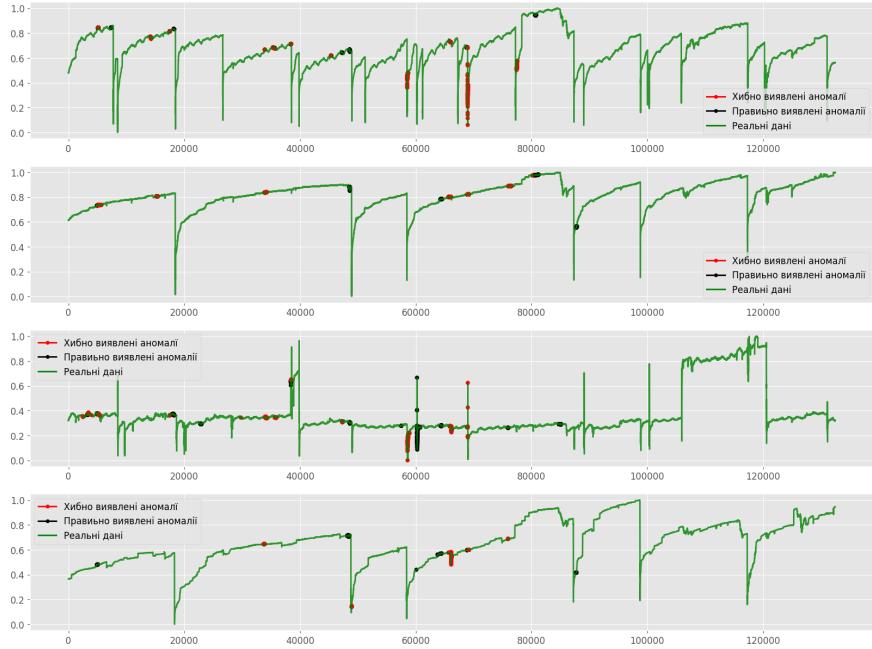


Рисунок 3.6 – Результат виявлення аномалій RNN SynLSTM Моделлю для PSM набору даних

Аналогічний результат для перших 4 ознак даних PSM.

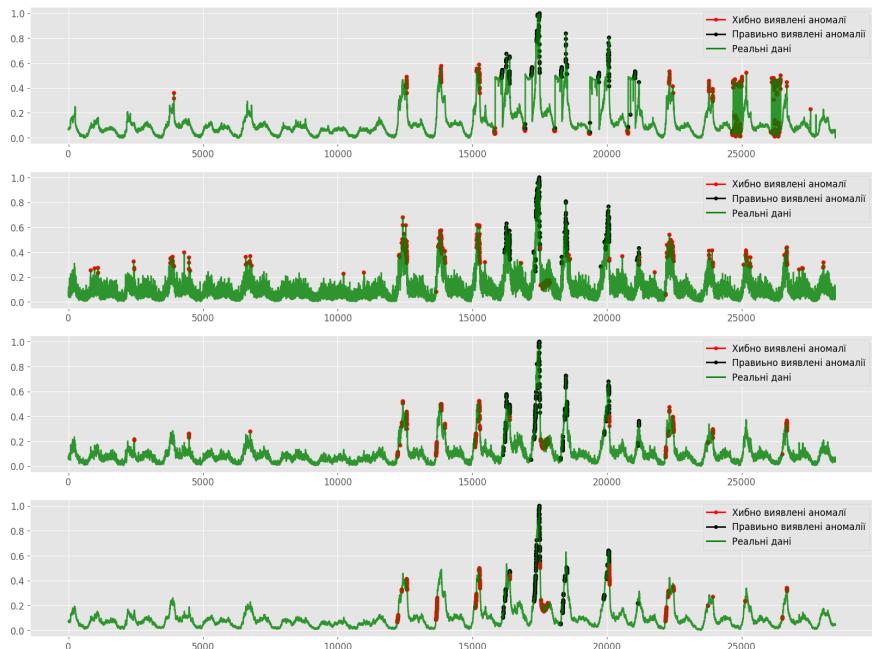


Рисунок 3.7 – Результат виявлення аномалій RNN LSTM Моделлю для SMD набору даних

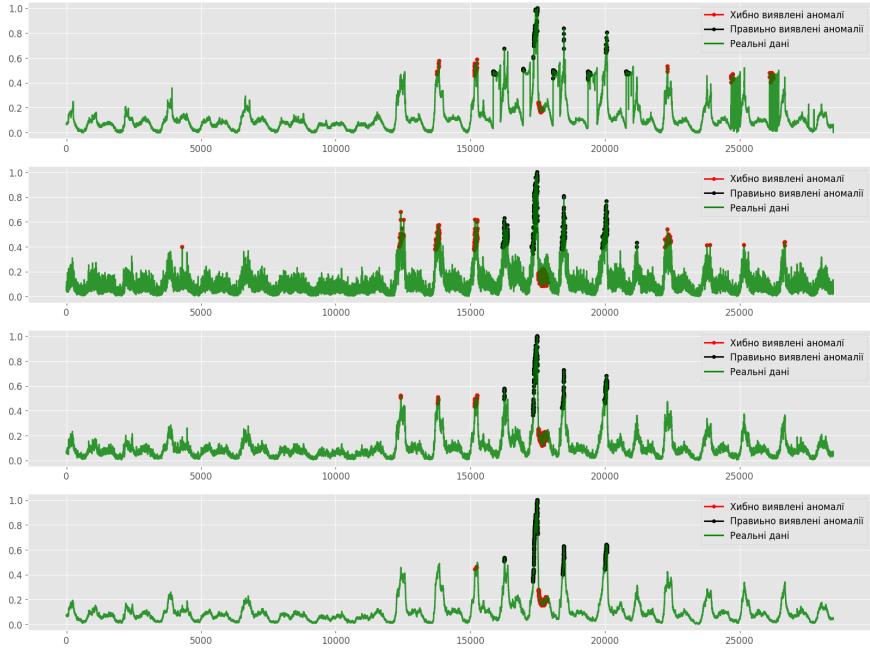


Рисунок 3.8 – Результат виявлення аномалій RNN SynLSTM Моделлю для SMD набору даних

3.3 Реалізація AE RANCoders та RANSynCoders моделей

Аналогічним чином до LSTM моделі виконується дослідження для RANCoders моделі. RANCoders на відміну від RANSynCoders моделі не має вбудованого шару, що відповідає за синхронізацію часових рядів. Для RANCoders та RANSynCoders моделей підбиралися однакові гіперпараметри.

1) N: Розмір вибірки для методу багаторазового заміщення (bootstrap) та кількість оцінювачів (estimators). Значення обчислюється як 5 помножене на округлене значення ділення ширини навчальної вибірки на 3, щоб отримати числа, кратні 5.

2) encoder_layers: Кількість прихованих шарів (hidden layers) для кожного енкодера.

3) decoder_layers: Кількість прихованих шарів для кожного декодера.

4) z: Розмір латентного простору (latent space). Обчислюється як

половина значення N, віднята одиниця.

5) activation: Функція активації, яка використовується для прихованих шарів моделі. У такій ситуації використовується функція ReLU (Rectified Linear Unit).

6) output_activation: Функція активації для вихідного шару моделі. У такому випадку використовується сигмоїдна функція.

7) S: Кількість компонент частоти, які використовуються для апроксимації вхідних сигналів.

8) delta: Параметр для методу заміщення даних (data augmentation).

9) batch_size: Розмір пакета (batch size), тобто кількість зразків, які обробляються за один крок навчання.

10) freq_warmup: Кількість епох попереднього навчання для синхронізації частоти.

11) sin_warmup: Кількість епох попереднього навчання для синхронізації сигналу.

12) epochs: Кількість епох навчання.

В Даній моделі 1 прихований кодер, 2 прихованих декодерів, використовується активація 'relu', вихідна активація сигмоїдна функція, delta = 0.05, розмір батча — 180, кількість епох — 10.

Ці параметри варіюються залежно від специфіки вашої моделі та задачі, яку ви намагаєтесь вирішити. Ви можете налаштувати ці значення відповідно до своїх потреб та експериментувати з ними для досягнення кращих результатів.

На наступних Рис. 3.9 та 3.10 можна побачити, те як дана модель виявляє аномалії в даних. На відміну від LSTM моделі яка прогнозує майбутні значення для ряду, ця модель будує довірчі інтервали (вони зображені сірим кольором). Якщо значення даних виходить за ці межі, то точку даних можна вважати аномальною.

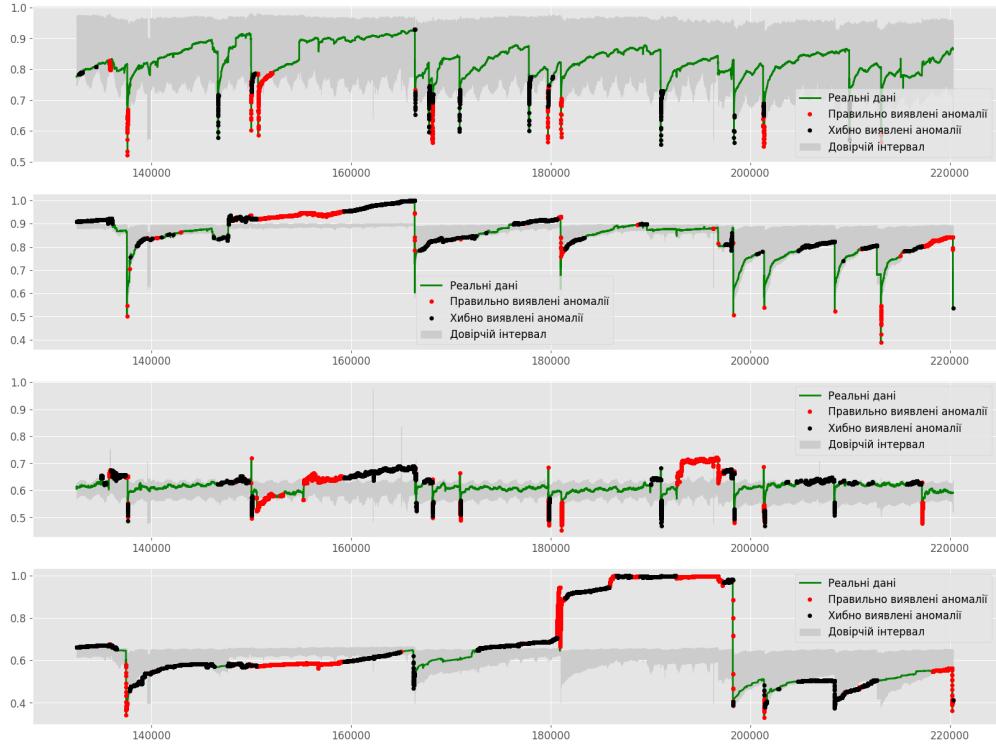


Рисунок 3.9 – Результат виявлення аномалій AE RANCoders Моделлю для PSM набору даних

Аналогічно для SMD набору даних. Як можна побачити з Рис. 3.10 модель виявляє велику кількість false positive помилок, в порівнянні з LSTM моделлю яка має меншу кількість false positive помилок.

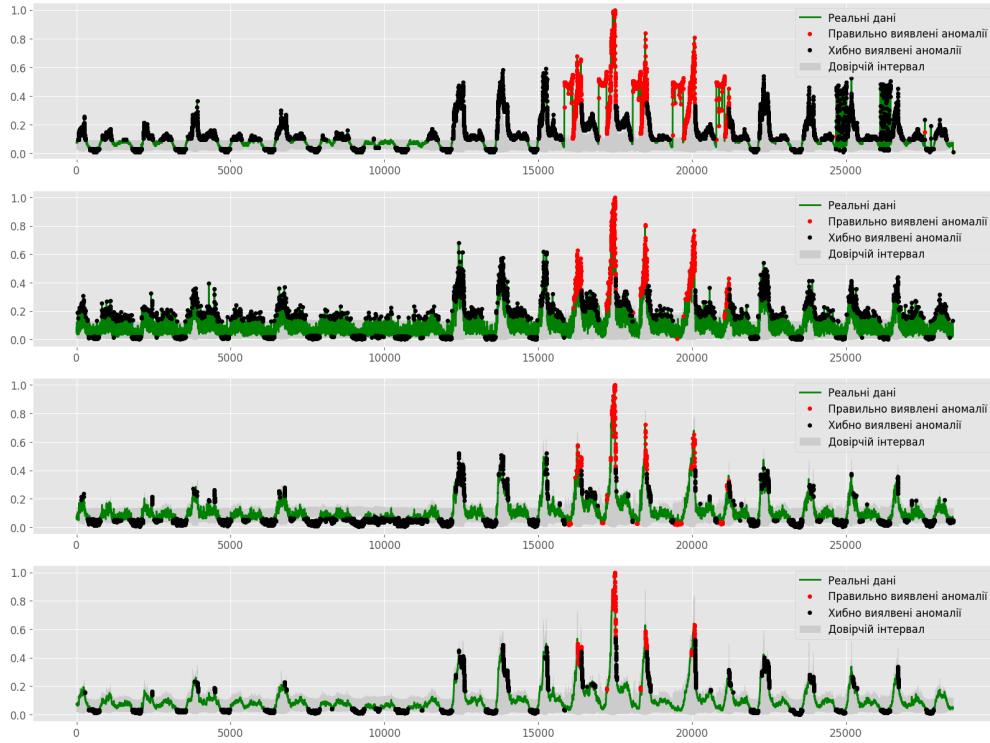


Рисунок 3.10 – Результат виявлення аномалій AE RANCoders Моделлю для SMD набору даних

Модель RANSynCoders навчається з тими самими гіперпараметрами що і RANCoders. Важливо зазначити, що синхронізація призводить до збільшення мультиколінеарності, таким чином зменшуючи кількість інформації та розмірність латентного простору. Цей факт можна чітко прослідкувати на Рис. 3.11 та 3.12.

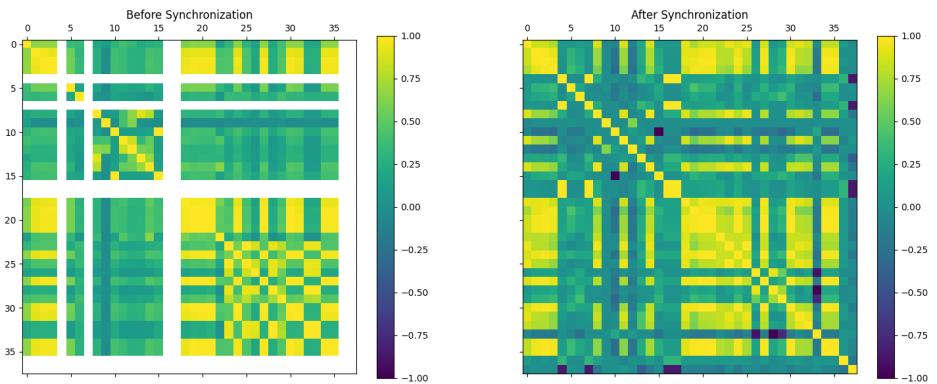


Рисунок 3.11 – Збільшення мультиколінеарності для SMD Датасету

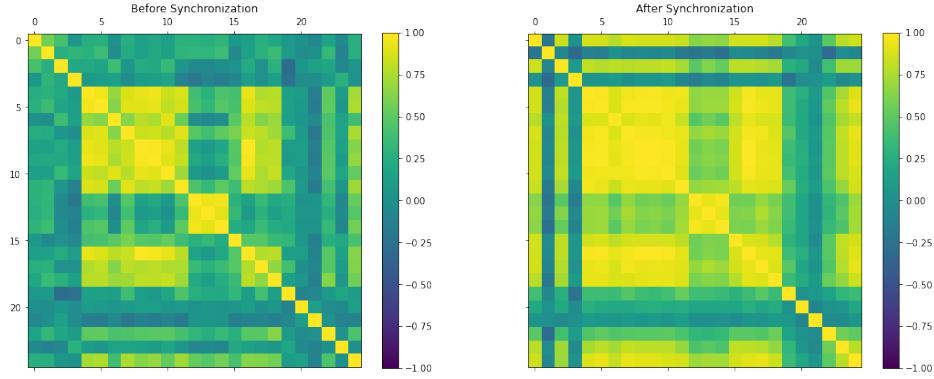


Рисунок 3.12 – Збільшення мультиколінеарності для PSM Датасету

Результати виявлення аномалій на Рис 3.13 та 3.14 моделлю RANSynCoders.

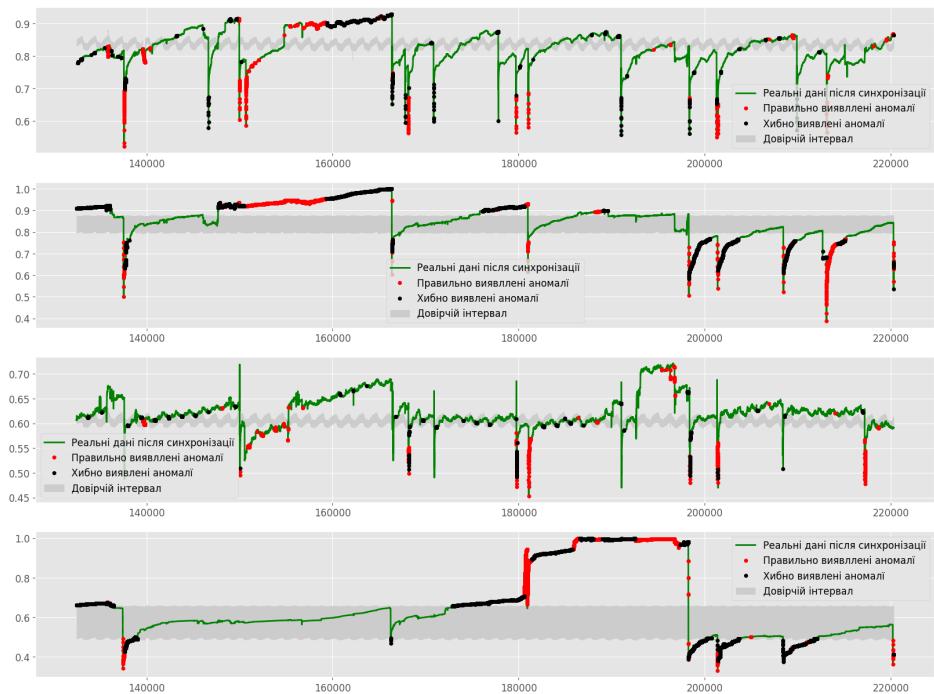


Рисунок 3.13 – Результат виявлення аномалій AE RANSynCoders Моделлю для PSM набору даних

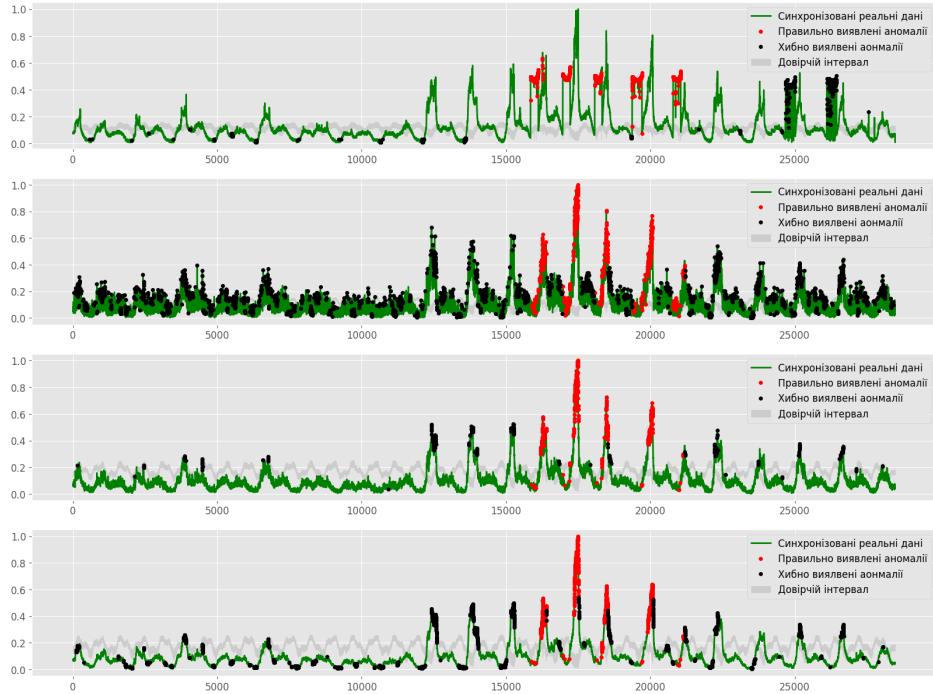


Рисунок 3.14 – Результат виявлення аномалій AE RANSynCoders
Моделлю для SMD набору даних

3.4 Порівняння результатів якості виявлення аномалій в даних

В наведеній таблиці 3.3 продемонстровано порівняння результатів виявлення аномалій в багатовимірних часових рядах для SMD та PSM датасетів.

Також в таблиці 3.4 можна оцінити у відсотках як змінилась F1-оцінка після додавання до моделі синхронізацію часових рядів. З таблиці видно, що після модифікації можна побачити дуже значне збільшення показників, або незначне погіршення.

Для порівняння результатів, використовуються метрики F1-показник, точність та повнота. Що показано на Рис. 3.15. Ці гістограми показують, що RANSynCoders в середньому краще впоралась з виявленням аномалій в запропонованих датасетах. Оцінки стабільно вищі.

Score	Model	SMD			PSM		
		F1	P	R	F1	P	R
F1-best	RNN LSTM	0.9887	0.9802	0.9974	0.9730	0.9634	0.9828
	RNN SynLSTM	0.9279	0.8676	0.9974	0.9391	0.8991	0.9828
	AE RANCoders	0.7667	0.6227	0.9974	0.9200	0.9090	0.9312
	AE RANSynCoders	0.9628	0.9561	0.9697	0.9009	0.8214	0.9974
F1-blind search	RNN LSTM	0.5108	0.3430	0.9999	0.8072	0.9100	0.7252
	RNN SynLSTM	0.5558	0.3849	0.9999	0.6577	0.9251	0.51031
	AE RANCoders	0.3274	0.1957	0.9999	0.3274	0.1957	0.999
	AE RANSynCoders	0.8310	0.7849	0.8828	0.8310	0.7849	0.8828
F1-twseq	RNN LSTM	0.3336	0.2002	0.9999	0.8751	0.8310	0.9241
	RNN SynLSTM	0.3845	0.2380	0.9999	0.84505	0.7702	0.9359
	AE RANCoders	0.4166	0.2631	0.9999	0.4541	0.2938	0.9994
	AE RANSynCoders	0.7904	0.9987	0.6539	0.9973	0.9999	0.9947
F1-seq	RNN LSTM	0.1734	0.0949	0.9999	0.4424	0.2842	0.9980
	RNN SynLSTM	0.1861	0.1026	0.9999	0.4588	0.3001	0.9739
	AE RANCoders	0.1728	0.0945	0.9999	0.4661	0.3039	0.9999
	AE RANSynCoders	0.7407	0.9959	0.5896	0.5104	0.3427	0.2003

Таблиця 3.3 – Порівняльна таблиця з метриками якості виявлення аномалій

Score	Model	% різниці оцінок	
		SMD	PSM
F1-best	RNN LSTM	-6.15	-3.48
	AE RANCoders	25.58	-2.08
F1-blind search	RNN LSTM	8.81	-18.52
	AE RANCoders	153.82	153.82
F1-twseq	RNN LSTM	15.26	-3.43
	AE RANCoders	89.73	119.62
F1-seq	RNN LSTM	7.32	3.71
	AE RANCoders	328.65	9.50

Таблиця 3.4 – Відсоткові зміни показників F1-міри після додавання до моделі синхронізації часових рядів

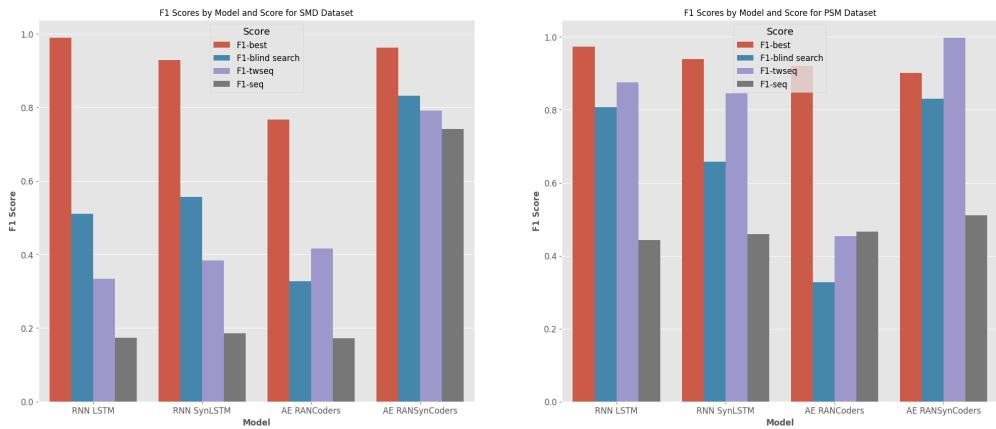


Рисунок 3.15 – Гістограма метрик якості виявлення аномалій для SMD та PSM датасетів

На наступному Рис. 3.16 зображене гістограму оцінок для кожної моделі в розрізі оцінок F1, Recall та Precision. Чорна вертикальна лінія демонструє дисперсію, тобто від яких і до яких меж ця оцінка варіюється в залежності від методології оцінювання.

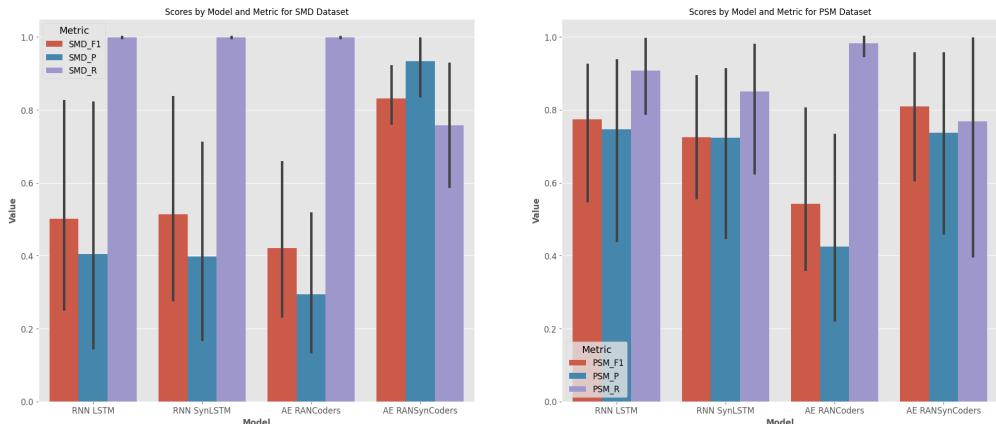


Рисунок 3.16 – Гістограма метрик якості виявлення аномалій в розрізі усереднених оцінок F1, Recall та Precision для SMD та PSM датасетів

Висновки до розділу 3

В цьому розділі були застосовані розглянуті у попередніх частинах моделі, алгоритми та метрики для задачі виявлення аномалій в

багатовимірних даних на практиці. Було використано два публічні датасети, що відповідають вимогам для виконання роботи. Обробка даних включала заповнення пропущених значень середніми значеннями та нормалізацію даних для підготовки їх для тренування моделі.

Для порівняння моделей глибинного навчання на основі RNN LSTM, AE RanCoders та RanSynCoders були використані датасети SMD (Server Machine Dataset) та PSM (Pooled Server Metrics). Обидва датасети містять багатовимірні дані та мають розмічену другу частину, яка включає аномалії.

Після обробки даних моделі були навчені та порівняні за допомогою відповідних метрик. Результати порівняння, що наведені в таблиці 3.3 дозволяють виявити сильні та слабкі сторони кожної моделі у виявленні аномалій в багатовимірних даних.

Варто зазначити, що додавання вбудованого шару, що відповідає за синхронізацію позитивно вплинув на метрики. це стосується як якості прогнозування часових рядів, для моделей на основі LSTM так і якості виявлення аномалій. Після синхронізації моделі демонструють значно кращі результати, якщо судити по F1-мірі приріст метрики може сягати до 328%, або мати погіршення що, в більшості випадків не перевищують 5%, що можна важати за допустиму похибку. Тобто моделі демонструють як мінімум результат не гірший ніж моделі без вбудованого шару з синхронізацією часових рядів.

Аналізуючи результати практичної реалізації моделей за Рис. 3.15 можна зазначити, що всі моделі змогли досягти дуже високої теоретичної оцінки для F1-міри, навідміну від практичної оцінки. Лише RanSynCoders показала дуже високі оцінки одночасно на двох датасетах для всіх методів оцінювання.

Також, з Рис. 3.16 можна зазначити, що RNN LSTM модель має стабільно високий рівень Recall, але низький рівень Precision з високою дисперсією, що свідчить про те, що результати моделі залежать від датасету на якому тестувалась модель. RanCoders має ще гірші

результати, але дисперсія не висока, що дає мотив думати, що модель працює не дуже добре, незалежно від даних. RanSynCoders модель має набагато кращі результати, ніж попередні моделі, що показує сталу ефективність для будь-якого датасету.

Отже, можна сказати, що отримані результати свідчать про те, що обрані моделі в повній мірі здатні виявлити аномальні дані в багатовимірних часових рядах серверних систем. Вбудований шар з синхронізацією часових рядів позитивно впливає на ефективність моделей. RanSynCoders є найкращою моделлю для виявлення аномалій в багатовимірних часових рядах, оскільки крім важливого високого показника Recall ця модель має стабільний високий Precision, така модель є збалансованою, що вигідно для будь-якого підприємства, оскільки це збереже ресурси компанії від затрат на опрацювання хибно позитивних сповіщень системи.

ВИСНОВКИ

В даній роботі було досліджено проблему виявлення аномалій в багатовимірних часових рядах та розроблено моделі LSTM на основі прогнозування часових рядів та реконструкції AE RAN Coders, та їх модифікації, що містять будований шар з синхронізацією часових рядів.

В Розділі 1 були наведені необхідні відомості про нейромережеві методи, що використовуються для виявлення аномалій. Було розглянуто підхід на основі прогнозування, де навчена модель використовується для прогнозування точки або послідовності на основі вхідних даних. Також були розглянуті AE моделі, що є простими у навченні та не вимагають складних обчислювальних операцій, що робить їх популярними у виявленні аномалій. Їх легко застосовувати до великих даних і вони здатні здійснювати вибір функцій для ефективного аналізу даних.

З цього розділу було отримане теоретичне підґрунтя про фундаментальні підходи до розв'язання задачі виявлення аномалій в багатовимірних часових рядах.

Кожна модель має свої переваги та недоліки. LSTM добре справляється з даними, які мають послідовний контекст. Ця модель ефективно працює з даними, де довгострокові залежності грають важливу роль. LSTM може моделювати складні залежності в часовихрядах, такі як сезонність, тренди, циклічність та інші нелінійні шаблони. AE слугує для витягування головних ознак з великого обсягу даних та зменшення їх розмірності. Він може бути корисним при аналізі даних з багатьма ознаками, де бажано знизити вимоги до обчислювальних ресурсів.

В Розділі 2 було розглянуто метрики для оцінки якості моделей. Запропоновано алгоритм сліпого пошуку порогового значення для досягнення максимального значення F1-оцінки в задачі виявлення аномалій та алгоритм F1best, що використовує метод brute-force, що

означає перебір усіх можливих порогів для виявлення аномалій з метою вибору оптимального порогу.

Як результат проведених досліджень, були отримані методології оцінювання, що в повному обсязі зможуть дати представлення про ефективність моделей.

Важливим елементом при оцінці ефективності будь-якої моделі є метрики. Вибір правильних метрик і методологій оцінювання має не меншу вагу, ніж сама побудова моделей. Тому розгляд проблеми вибору метрик заслуговує детального підходу, щоб отримати повну інформацію про ефективність моделей.

В Розділі 3 були реалізовані моделі LSTM, RANCoders та їх модифікації SynLSTM Та RANSynCoders. Детально проаналізовано результати метрик.

Як результат, можна відзначити що, додавання вбудованого шару, відповідального за синхронізацію, має позитивний вплив на метрики як у випадку прогнозування часових рядів для моделей на основі LSTM, так і у виявленні аномалій. Після синхронізації моделі показують значно кращі результати. Наприклад, за F1-мерою приріст метрики може становити до 328%, або в окремих випадках може бути незначним зниженням, яке, в більшості випадків, не перевищує 5%. Такі зміни можна вважати прийнятною похибкою. Загалом, моделі з вбудованим шаром синхронізації часових рядів демонструють принаймні такі ж результати, як моделі без такого шару.

Отримані результати свідчать, що обрані моделі повністю здатні виявляти аномальні дані в багатовимірних часових рядах серверних систем. Вбудований шар синхронізації позитивно впливає на ефективність моделей. Зокрема, модель RanSynCoders виявляє аномалії в багатовимірних часових рядах найкраще, оскільки, крім високого Precision, вона має стабільно високий Recall. Така збалансованість моделі є вигідною для будь-якого підприємства, оскільки дозволяє зберегти ресурси компанії, уникнувши витрат на обробку помилково позитивних

повідомлень системи.

Подальшими кроками для дослідження в цій темі може бути оптимізація гіперпараметрів, застосування ансамблевих моделей, які поєднують кілька незалежних моделей для досягнення кращої загальної ефективності, вдосконалення алгоритмів для пошуку порогового значення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Deep Learning for Anomaly Detection: A Survey / Raghavendra Chalapathy University of Sydney, Capital Markets Cooperative Research Centre, Sanjay Chawla Qatar Computing Research Institute, Hbku. — 2019.
2. Forecasting of COVID-19 using deep layer recurrent neural networks (RNNs) with gated recurrent units (GRUs) and long short-term memory (LSTM) cells / K. E. ArunKumar, D. V. Kalaga, C. M. S. Kumar et al. // Chaos, Solitons Fractals. — 2021.
3. Krizhevsky, A. Imagenet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G.E. Hinton // Advances in neural information processing systems. — 2012. — Pp. 1097–1105.
4. Golestani, Abbas. Can we predict the unpredictable? / Abbas Golestani, Robin Gras // Scientific Reports. — 2014. — Vol. 4, no. 1. — Pp. 1–6.
5. Long short term memory networks for anomaly detection in time series / Pankaj Malhotra, Lovekesh Vig, Gautam Shroff et al. // Proceedings of ESANN. — Vol. 89. — 2015. — Pp. 89–94.
6. Provotar, O. I. Unsupervised anomaly detection in time series using LSTM-based autoencoders / O. I. Provotar, Y. M. Linder, M. M. Veres // IEEE International Conference on Advanced Trends in Information Theory (ATIT). — Kyiv, Ukraine: 2019. — Pp. 513–517.
7. Network anomaly detection using LSTM based autoencoder / M. S. Elsayed, N. A. Le-Khac, S. Dev, A. D. Jurcut // Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet '20). — New York, NY, USA: 2020. — Pp. 37–45.

8. Real-time anomaly detection for flight testing using autoencoder and LSTM / Z. Que, Y. Liu, C. Guo et al. // International Conference on Field-Programmable Technology (ICFPT). — Tianjin, China: 2019. — Pp. 379–382.
9. Abdulaal, Ahmed. Practical Approach to Asynchronous Multivariate Time Series Anomaly Detection and Localization / Ahmed Abdulaal, Zhiyuan Liu, Tomasz Lancewicki // Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. — 2021. — August 14-18.
10. Breiman, Leo. Bagging predictors / Leo Breiman // Machine Learning. — 1996. — Vol. 24, no. 2. — Pp. 123–140.
11. An Efficient and Scalable Deep Learning Approach for Road Damage Detection / Sadra Naddaf-Sh, M-Mahdi Naddaf-Sh, Amir R. Kashani, Hassan Zargarzadeh // 2020 IEEE International Conference on Big Data (Big Data). — 2020. — Pp. 5602–5608.
12. Provotor, Oleksandr I. Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders / Oleksandr I. Provotor, Yaroslav M. Linder, Maksym M. Veres // 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT). — 2019. — Pp. 513–517.
13. DeCastro-García, Noemí. Machine learning for automatic assignment of the severity of cybersecurity events / Noemí DeCastro-García, Ángel L Muñoz Castañeda, Mario Fernández-Rodríguez // Computational and Mathematical Methods. — 2020. — Vol. 2, no. 1. — P. e1072.
14. Gouliaras, Dionysis. Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data / Dionysis Gouliaras, Sani Kamis // 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML). — 2019. — Pp. 12–17.
15. USAD: UnSupervised Anomaly Detection on Multivariate Time Series / Julien Audibert, Pietro Michiardi, Frédéric Guyard et al. //

Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. — 2020. — Pp. 3395–3404.

16. Robust anomaly detection for multivariate time series through stochastic recurrent neural network / Ya Su, Youjian Zhao, Chenhao Niu et al. // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. — 2019. — Pp. 2828–2837.